

Copyright

by

Rahul Rai

2006

**The Dissertation Committee for Rahul Rai Certifies that this is the approved
version of the following dissertation:**

QUALITATIVE AND QUANTITATIVE SEQUENTIAL SAMPLING

Committee:

Matthew I. Campbell, Supervisor

Richard Crawford

Carolyn C. Seepersad

Thomas Sager

Dewayne E. Perry

QUALITATIVE AND QUANTITATIVE SEQUENTIAL SAMPLING

by

Rahul Rai B.Tech.; M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August, 2006

Dedication

To my family, friends, and my mentors.

Acknowledgements

First and foremost, I thank my advisor, Matthew Campbell, from whom I learned so much, for being my mentor and teacher; a never-ending source of inspiration, encouragement, and support – even through the most trying of times. It has been an invaluable experience to study with him, and I will always feel fortunate and honored to have had the opportunity to be his *first* PhD. student from the Automated Design Lab (ADLab).

Special thanks also goes to the members of my thesis committee, who have all taken the place of mentors and teachers to me; Richard Crawford, Carolyn Seepersad, Thomas Sager, and Dewayne Perry.

Further, I thank everyone who was or is in the Automated Design Laboratory, MADLab, and SFF Lab at UT-Austin for working with me and being my friends.

I would also like to acknowledge my family and thank them; my mother, Shanti, for her unconditional love and support, my father, Vidya, for love and motivation also, all these years, and all my siblings: Neelam, and Varun, for being their delightful selves. Motivational and caring support of my wife Shraddha during the final stages of my dissertation is worth special thanks.

And, of course, thanks to my friends in UT Racquetball club and Austin Dart league, for giving me confidence and support and for believing in me, and relaxing me with good extracurricular activities.

Last but not least, sincere thanks to my undergraduate advisor Dr. Manoj Kumar Tiwari for showing me the door to the research world.

Many Thanks
Rahul Rai, August 2006

QUALITATIVE AND QUANTITATIVE SEQUENTIAL SAMPLING

Publication No. _____

Rahul Rai, Ph.D.

The University of Texas at Austin, 2006

Supervisor: Matthew I. Campbell

Sequential sampling refers to a set of design of experiment (DOE) method where the next sample point is determined by information from previous experiments. This dissertation introduces qualitative and quantitative sequential sampling (Q2S2) technique, in which optimization and user knowledge is used to guide the efficient choice of sample points. This method combines information from multiple fidelity sources including computer simulation models of the product, first principals involved in design and designer's qualitative intuitions about the design. Both *quantitative* and *qualitative* information from varying fidelity sources are merged together to arrive at new sampling strategy. This is accomplished by introducing the concept of confidence function, C , which is represented as a field that is a function of the decision variables, \mathbf{x} , and the performance parameter, f . The advantages of the approach are demonstrated using various function example cases. The examples include design of a bi-stable Micro Electro Mechanical System (MEMS) relay, a complex and relevant mechanical system. In each case, the performance of Q2S2 is highly encouraging.

List of Tables	x
List of Figures	xi
CHAPTER 1 INTRODUCTION AND MOTIVATION	1
1.1. Design as a Discipline.....	1
1.2. The Problem: Systems Design	3
1.3. Pushing the Envelop	5
1.3.1. Developing Empirical Models	6
1.3.2. Developing Mechanistic Models	6
1.3.3. Empirical versus Mechanistic Models	7
1.4. Design Space Exploration and Representation	8
1.4.1. Exploration of the Design Space.....	9
1.4.2. Representation of the Design Space	9
1.4.3. Role of Human Designer	10
CHAPTER 2 BACKGROUND	13
2.1. Design of Experiments.....	13
2.1.1. Introduction to Some Experimental Design Terminology	15
2.1.2. Random and Quasi-random Methods	17
2.2. Sampling Techniques.....	17
2.2.1. Full Factorial (FF).....	18
2.2.2. Orthogonal Array (OA).....	18
2.2.3. Latin Hypercube Design (LHD)	19
2.2.4. Hammersley Sequential Sampling (HSS).....	19
2.3. Adaptive Sampling For Optimization	20
2.3.1. Sequential Sampling Criteria	20
2.4. Metamodels.....	23
2.4.1. Geometric Metamodels: Response Surface Models	25
2.4.2. Geometric Metamodels: Spline-Based Metamodels.....	27

2.4.3. Cubic Splines	27
2.4.4. MARS: Multivariate Adaptive Regression Splines	28
2.4.5. NURBs-based Metamodels.....	30
2.4.6. Stochastic Metamodels: Kriging Models.....	32
2.4.7. Stochastic Metamodels: Radial Basis Function Models	34
2.4.8. Heuristic Metamodels: Neural Network Models	36
2.4.9. Heuristic Metamodels: Support Vector Machines.....	38
CHAPTER 3 RESEARCH QUESTIONS AND HYPOTHESIS	41
3.1. Design Problems Classification	41
3.2. Illustrative Example and Definitions	45
3.3. Research Questions.....	47
3.4. Hypothesis.....	48
CHAPTER 4 FORMULATION OF QUANTITATIVE AND QUALITATIVE SEQUENTIAL SAMPLING (Q2S2) PROCESS	50
4.1. Confidence as a Function of x and f	50
4.2. Qualitative and Quantitative Sequential Sampling (Q2S2)	58
4.2.1. Merging Functions	62
4.2.2. Mirroring Function of C - f - x plots.....	64
4.3. Analytical Expression Derivation	65
CHAPTER 5 IMPLEMENTATION EXAMPLE FOR KNOWN FUNCTIONS	68
5.1. The Single-Variable Functions	68
5.1.1. Example Function 1	68
5.1.2. Example Function 2	75
5.2. Multiple-Variable Functions and Comparison with Other Sampling Techniques	78
5.2.1. Metamodel Accuracy Measures.....	79
5.2.2. Sample Size and Metamodel.....	80

5.2.3. Multiple-Variable Functions	81
5.2.4. Discontinuous Function	86
5.3. Tuning the knobs	88
5.4. Recapitulation	92
CHAPTER 6 Q2S2 IMPLEMENTATION FOR DESIGN OF BI-STABLE MEMS RELAY	93
6.1. Introductory Comments and Motivation Behind Bi-Stable MEMS Relay	93
6.1.1. Multistable Equilibrium Systems	93
6.1.2. Micro-Electro Mechanical (MEMS) Relays	94
6.1.3. Understanding Characteristics of MSE Systems	95
6.1.4. Representation of Complaint Structures	97
6.2. Q2S2 Based Solution Approach	98
6.3. Recapitulation	102
CHAPTER 7 CONCLUSIONS	104
7.1. Summary of Results	104
7.2. Reprise of Hypothesis	106
7.3. Conclusions on Why the Hypothesis Is Supported	106
7.4. Suggestions for Future Research	108
Appendix	111
Software Manual	111
Framework and Important Files	111
The Subroutines	113
References	122
Vita	132

List of Tables

Table 2.1 Different switching criteria (taken from [Sasena 200b])	22
Table 4.1 Different functions for representing C fields	55
Table 5.1 Multivariable Test Problems	81
Table 5.2 Assumed Monotonicity Information for the Test Problems	81
Table 5.3 Bi-stable structure design variables and associated monotonicity information.....	90
Table 6.1 Bi-stable structure design variables and associated monotonicity information.....	100

List of Figures

Figure 1.1 Expansion of the Design Space	5
Figure 1.2 Metamodel for an expensive analysis.....	10
Figure 1.3 Designer use knowledge from many sources to design a product.....	11
Figure 2.1 Metamodel Concept. Metamodels can be thought of as a representation of a system of models (taken from Cameron 2005).....	23
Figure 2.2 Neural Network based metamodeling	36
Figure 4.1 A single experiment in the design of a heat sink shows an experiment run at $l = 4$ in. yields a surface temperature of 600°F	51
Figure 4.2 The C-field created by introducing the results of a single experiment.....	52
Figure 4.3 The C-field that results from including monotonicity information	53
Figure 4.4: The Q2S2 process is an cycle of six main steps.....	59
Figure 5.1 One variable example function 1	68
Figure 5.2 Metamodel after 2 sample points	69
Figure 5.3 C field for the first sample point for example function 1	70
Figure 5.4 C field for the second sample point for example function 1	70
Figure 5.5 The C' field for the first sample point for the example function 1	71
Figure 5.6 The C' field for the second sample point for the example function 1	72
Figure 5.7 The C' field after merging and mirroring information from two sample point for the example function 1 and location of minima for the plot.....	72
Figure 5.8 The metamodel after 3 sample points.....	73
Figure 5.9 The step 5 through 7 of the Q2S2 process for adding 5 th sample point	74
Figure 5.10 Sequentially sampling for example 1: a) after six points the accuracy is 95%, b) after ten points the accuracy is over 99%.....	75

Figure 5.11 One Variable example function 2 taken from Martin and Simpson.....	76
Figure 5.12 The dialog between the Q2S2 process and the user currently takes place through a Matlab command window	76
Figure 5.13 Sample points, actual grey box model and metamodel predicted by Q2S2 for example function 2.....	78
Figure 5.14 Box plots for multivariable test function 1	83
Figure 5.15 Box plots for multivariable test function 2.....	83
Figure 5.16 Box plots for multivariable test function 3.....	84
Figure 5.17 Box plots for multivariable test function 4.....	84
Figure 5.18 Box plots for multivariable test function 5.....	85
Figure 5.19 Heaviside step function (a) actual plot (b)oscilloscope plot	86
Figure 5.20 The metamodel and sample points for Heaviside step function after 11 sample points in absence of any qualitative information.....	87
Figure 5.21 The metamodel and sample points for Heaviside step function after 11 sample points in presence of first set of qualitative information	87
Figure 5.22 The metamodel and sample points for Heaviside step function after 11 sample in presence of second set of qualitative information	88
Figure 6.1 A) Potential Energy as a function of changing configuration of a MSE system B) Force vs Displacement curve of a MSE system [King, 2004]...	96
Figure 6.2 Step-by-step decoding of the representation of the structure: a) First beam of length l_1 is attached at the origin at orientation angle θ_1 ; b) Second beam of length l_2 and orientation angle θ_2 is added to the end of the first beam; c) The symmetrical half of the structure after adding all the beams and anchoring the beam ends; d) The final structure. Also, note that the origin is considered as the loading point.	98

Figure 6.3 SLS manufactured macro prototypes.....	101
Figure 6.4 Feasible solution 1 generated in ANSYS by Q2S2 Methodology	101
Figure 6.5 Feasible solution 2 generated in ANSYS by Q2S2 Methodology	102
Figure A.1 Organization of MATLAB code for Q2S2.....	112

CHAPTER 1

Introduction and Motivation

The field of engineering design is as unique a discipline as it is challenging. In the past, it has undergone changes in ideation, method, and manifestation, ranging from a redefinition of methodologies to entire paradigm shifts. With recent advances in computational power and resources, as well as input from traditionally separate fields, the range of possibilities continues to grow and open up further areas of potential improvements to the design process.

1.1. Design as a Discipline

Traditionally, the design of complex mechanical systems such as automobiles has been regarded more as a craft or an art than an engineering discipline. The consensus was that a good designer had to be an expert in many fields, with excellent intuition, talent, and a feel for how to compose concepts that would prove viable given a set of customer requirements. It was thought that design expertise could not necessarily be *taught*, but that it had to be *developed* through experience and practice, and the designers that emerged from this culture were and are held in high esteem by the engineering community.

This approach still carries validity, and there are few no methods that can replace the engineering intuition of an experienced designer. However, this approach is now being augmented by methods and ideas from fields concerned with information and uncertainty, such as mathematics, as well as new emerging sciences through the enabling abilities of more advanced computers. Design automation methods and tools are created to aid the designer, as the design of complex systems becomes more complex and

challenging. As such, engineering design is experiencing a new range of possibilities, which require investigation and research in their own right, turning a craft into a new engineering discipline of its own.

As research in optimization advanced and computational resources improved, designers welcomed the idea of striving toward an optimal design rather than a merely feasible one. However, this effort carried an inherent handicap, because the problem was actually not well defined: what was the space they were optimizing over? What were the constraints? How truly optimal is a design given limited evaluation within a computer?

Another problem was that design was often considered to be the conglomerate of many different disciplines and not a discipline in itself. Thus, the immense freedom that has been gained from the introduction of computers has created possibilities for design rarely utilized previously. To this day, some design projects are still conducted in suboptimal ways because the full potential of design as a discipline has not been implemented or even recognized.

Essentially, the problem of designing complex and complicated systems has shifted since the advent of improved computational resources from the creation of data, to the selection of what particular data to create and upon which to focus. It has become an information management problem, not unlike those found in the field of computer science. One of the most challenging decisions a designer is faced with today is the question of how to assess and process the information available to him. Additionally, he/she needs to make appropriate decisions about where more information should be gathered and the quality of information that this requires.

Engineering design in general, and systems design in particular, are thus relatively new and emerging disciplines in terms of mathematical and methodological rigor; especially as compared to other disciplinary fields of engineering. These areas are

substantially enabled by and also linked to advances in computer science and related fields. They range from the development of new design methodologies and methods to specific techniques employing uncertainty, probabilistic and robust design as well as procedures to facilitate the assessment of technological advances from which the system could benefit.

1.2. The Problem: Systems Design

When exploring the concept of complexity in systems design, there are three separate issues that emerge as particularly relevant and challenging to this field:

1. Different disciplines with a *high degree of expertise and specialized disciplinary knowledge* need to be merged and accounted for;
2. *High functional interconnectivity* exists between the disciplines; there are cross-disciplinary effects on the same parts of the design and interconnection in the functional processes effecting outputs as well as inputs;
3. *Stringent goals, requirements and constraints* are imposed on the design at the system and disciplinary levels;

While the presence of different disciplines is concurrent in nearly all types of design tasks, when dealing with complex systems such as aircraft, and cars, this aspect becomes dominant. In such a highly evolved and technical system each of the disciplines represents a science in itself, and contains an entire body of research and development work. Thus, there are high levels of disciplinary expertise that cannot be ignored when analyzing or designing such a system. At the same time, the sheer number of disciplines and the depth of the knowledge in each prohibit any single designer to represent them accurately at the systems level so as to do them all justice.

Secondly, within a complex system, these different disciplines are interconnected to a very high degree – influencing each others inputs and outputs – and often affecting the same or closely related parts of the design configuration itself. The result is that there is no simple breakdown of the disciplines, and none of them can be evaluated in isolation. It is this aspect of interaction and interconnectivity between the disciplines that makes current products *complex* in nature. This is particularly apparent when a physics-based approach is employed instead of an industrial data based approach, since the interactions among disciplines stem from the actual physics of the problem.

Finally, the demands and requirements that are typically placed on product design itself have become far more demanding recently than in the past. The reasons for this are multifold, the predominant being budget constraints now imposed which drive commercial product design. This factor, coupled with expanded enthusiasm about new technologies and the many problems they should be able to solve, have placed the system s design team s in a position where they find they have to live up to decidedly high expectations. It is this aspect of more restrictive budgets and higher demands that has lead to the paradigm shift in engineering design which is underway.

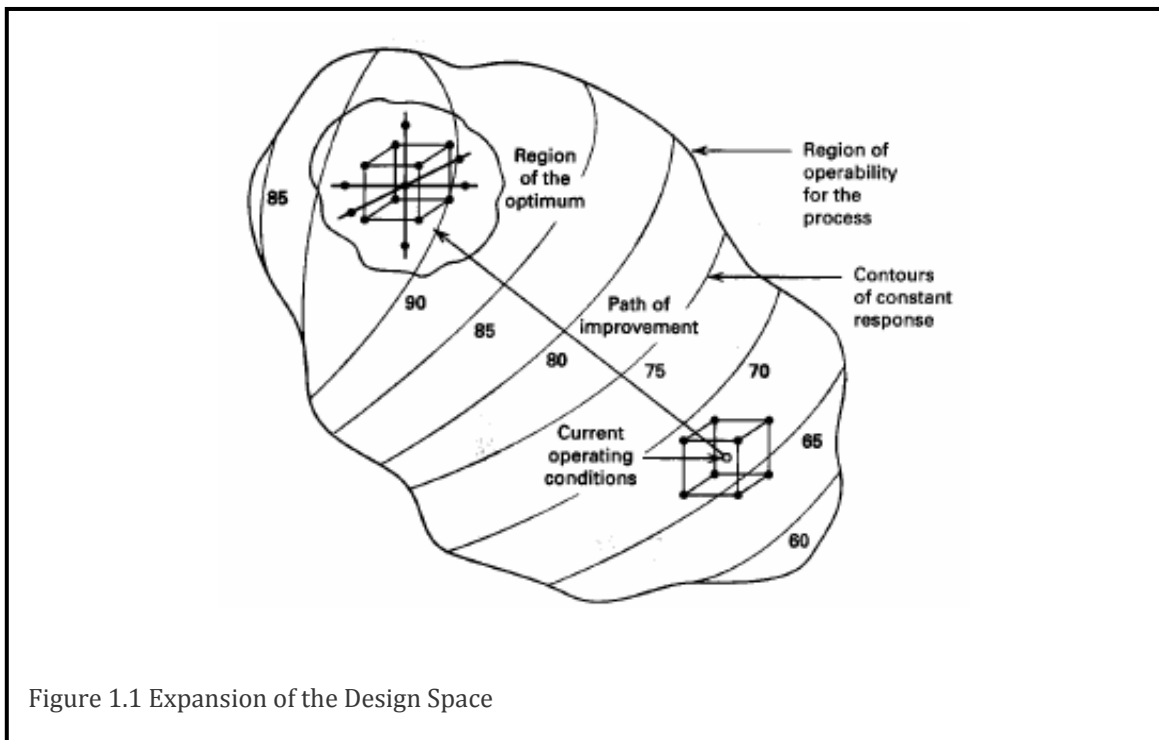
The combination of these three issues poses interesting and demanding information management problem for system level design. The high interconnectivity of disciplines combined with the specific knowledge and expertise in each of these requires a systems level view that is broad, i.e. can span across all disciplines, while at the same time remains detailed enough to do the disciplinary knowledge justice. The treatment of these problems requires highly evolved information management and decision approaches, which result in design methodologies that can handle this high degree of complexity.

1.3. Pushing the Envelop

In order to utilize the full potential of design methodologies it is important to realize that better solutions might lie outside the familiar space – in the space of all possible configurations, beyond any explored so far. This thought has become known as *pushing the envelop in design space*.

Before beginning a detailed explanation on how this is done, it is beneficial to step back and look at the big picture: Why push the design envelop at all? If similar past designs were sufficient for this purpose, then why create new concepts? The answer is simple: by pushing the design envelop we can arrive at design solutions that may perform better than conventional ones.

The reason conventional configurations were usually selected in the past was not



because they were thought to be the *best* possible configurations, but because they were known to *work*; and improvements were sought only within a small neighborhood of the familiar territory [Figure 1.1]. This raises the question of how to assess a revolutionary or otherwise unconventional configuration *accurately* and *quickly* in the theoretical realm. The answer to this question lies in the root of developing design models.

There are 2 main approaches to developing design models:

1. Empirical or data based modeling.
2. Building models based on the underlying physics and chemistry governing the behavior of the process.

1.3.1. Developing Empirical Models

1. Collect data from the process
2. Specify the correlation structure between variables, e.g. polynomials; time-series; artificial neural networks
3. Use a numerical technique to find parameters for the structure such that the correlation between the data is maximized
4. Validate the model against an 'unseen' data set
5. If model is not satisfactory, go to step 2.

Thus, we can see that in empirical modelling much depends on the availability of representative data for model building and validation. Furthermore a trial and error approach such as this requires little in terms of process knowledge apart from cause-and-effect between variables.

1.3.2. Developing Mechanistic Models

The development of mechanistic models follows a different procedure:

1. Use fundamental knowledge of the interactions between process variables to define the model structure

2. Perform experiments to determine the parameters of the model
3. Collect data from the process to validate the model
4. If the model is not satisfactory, go to step 1 and re-examine process knowledge

Mechanistic modelling therefore does not require much data for model development, and hence is not subject to the idiosyncrasies in data. It requires a fundamental understanding of the physics governing the problem. Usually the mechanistic modelling process can be very time consuming.

1.3.3. Empirical versus Mechanistic Models

When available, mechanistic models can provide more realistic and useful predictions than experimental analyses. For example, the details contained within a mechanistic model offer the opportunity to test the sensitivities of the process to meaningful entities such as heat transfer coefficients; activation energies; catalyst poisoning, etc. With very few exceptions, the parameters of data based models are just numbers encapsulating combined effects. Thus, it is very difficult to attach physical meaning to them, and hence such sensitivity studies cannot be performed.

While mechanistic models can be used to explore the design space, empirical models are frequently used as the bases for design improvement in the near by space of existing design. The argument here is that empirical model based design only require the models to represent with some accuracy. Nevertheless, if a mechanistic model is available, it will be foolish to discard it in favor of a data based one.

Another comparison that is always made between the two modelling approaches is that of cost. Due to the complexity of many processes, mechanistic modelling is indeed very expensive in terms of human effort and expertise. Employing physics-based mechanistic models rather than empirical, database driven analysis implies a greater need

for computation and higher fidelity tools. This quickly creates a combinatorial problem of computationally expensive function calls when aspiring to assess a reasonably large space. While this may be acceptable if there is only a single configuration or point design to be analyzed, it becomes prohibitive when aspiring to explore an entire design *space*. Hence, there is a definite need for empirical approximation models which provide predictions of selected responses with satisfactory precision and reliable accuracy while requiring a minimum of data points to be sampled from the space in the interest of efficiency. The advantage with empirical modelling lies in the fact that empirical modelling will deliver some form of working model in a much shorter time. It requires large amounts of representative data, and in many instances, these can only be acquired by perturbing the process via planned experiments. The concept of Design of Experiment (DOE) can help by creating empirical approximation models of the physics-based analyses and thus reducing computational costs while artificially maintaining the required fidelity and providing us a mean to push the design envelope. The field of study, sometimes referred to "Design of Experiments" (or DOE) deals with what efficient set of experiments should be run to best understand a given phenomena. DOE and in particular Metamodeling comprise a group of statistical techniques for empirical model building and analysis, as is described later in this chapter.

1.4. Design Space Exploration and Representation

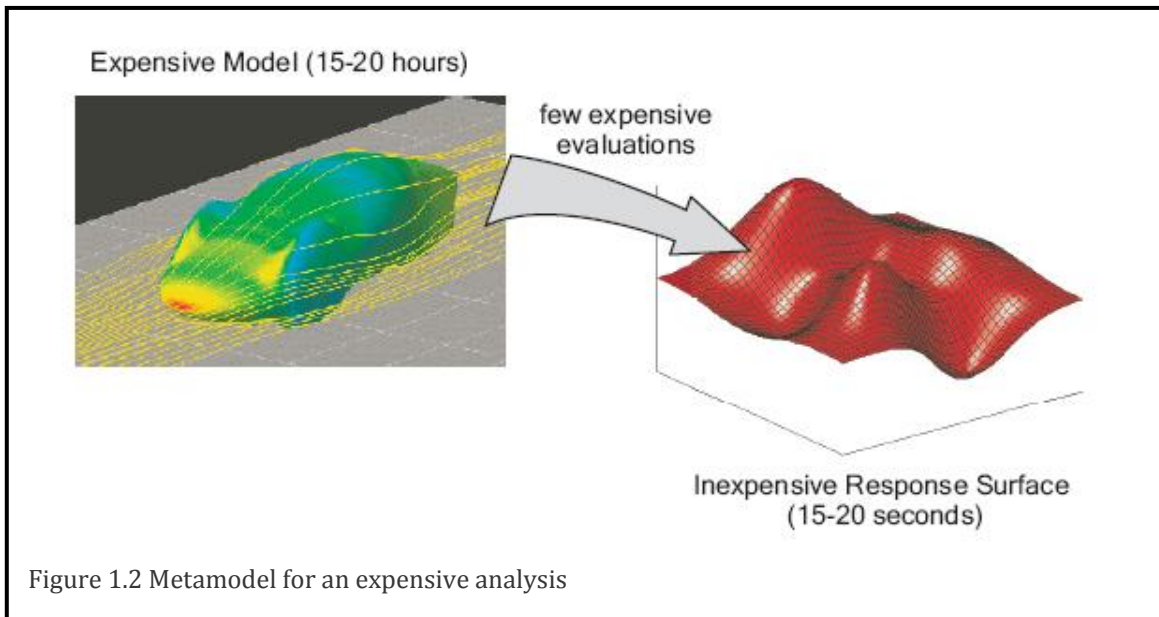
In conceptualizing the design space, two main challenges must be overcome. First, the design space must be *explored* in planned manner so that the empirical model building process may characterize the design space based upon a collected data set. Second, the resulting data must be characterized with an efficient *representation* of the design space in the form of a metamodel.

1.4.1. Exploration of the Design Space

Exhaustively collecting experimental data or sample points is not a cost effective solution to the need to search the design space. The underlying goal of a set of experiments or simulations is to develop information about the behavior of the unknown design space that determines the response of the experiment or simulation. Ideally, a sufficient number of simulations or experiments will be performed to clearly define the behavior of the design space. But in many cases, experiments and simulations can provide only limited information about the behavior of the design spaces that they are intended to survey because of the cost of experimentation or simulation. Since only restricted amounts of data are realistically available from experimental sources, simulations and mechanistic models it is crucial that any data collected must be collected efficiently. Classical Design of Experiments (DOE) techniques augmented with Sequential Sampling Techniques (SST) offer methods to improve data collection efficiency, as discussed in Chapter 2.

1.4.2. Representation of the Design Space

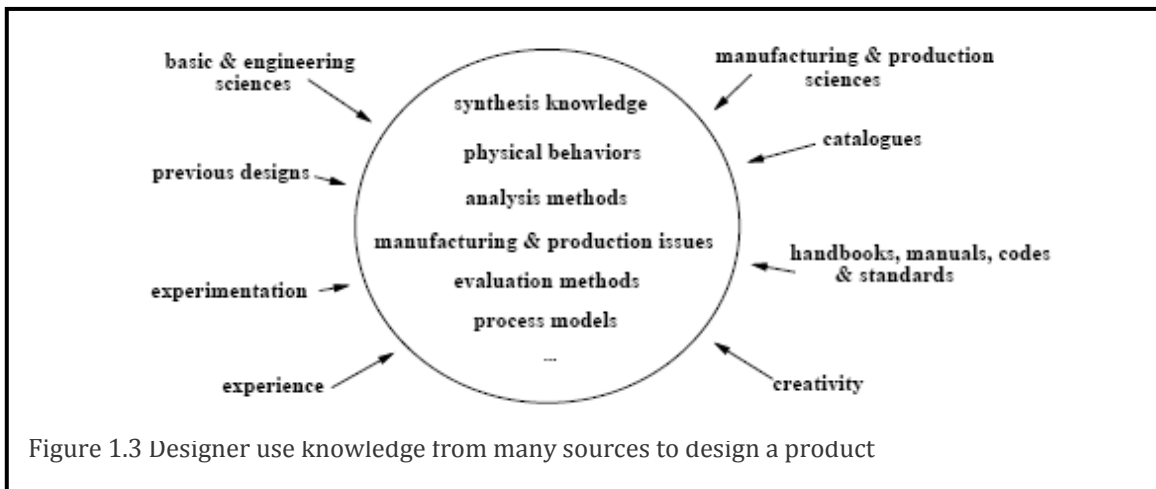
Once an efficient data set has been collected, the next challenge is to build an adequate data representation or metamodel. Metamodels, models of models, are a rapidly emerging engineering technique that offers great potential to improve engineering design. As engineering design becomes increasingly data driven, with reduced cycle times, fewer prototypes, and increasingly complex, multidisciplinary system designs, the task of identifying and uncovering fundamental relationships for a design becomes important. A more pragmatic, yet highly effective approach is that of building approximate models of computationally expensive analysis codes, which has gained wide acceptance in the design community. These approximate models (Figure 1.2) are computationally benign and allow the designer to comprehensively explore the design space. Metamodels are



valuable for this application. *Metamodels*, like other models, are also created based on data. In the case of metamodels, however, the data collected is produced from analysis codes which are run for multiple configurations. The word metamodel is derived from *meta*, a Greek presyllable which means 'over' or 'above'. In essence, metamodels are actually similar to the mathematical models, which are developed and used in science to describe physical phenomena. They are necessary for representation of design space because the level at which design analysis takes place is so far removed from the physics-based mechanistic models that the complexity of the design problem becomes too intense to use physics-based mechanistic models directly.

1.4.3. Role of Human Designer

In designing a product, designers bring together knowledge from various sources for use in context as shown in Figure 1.3. The sources may include basic sciences such as physics and chemistry, engineering sciences such as thermodynamics and fluid mechanics, manufacturing and production sciences, empirical knowledge from handbooks and manuals, catalogs, previous designs, experience and creativity of



designers, *etc.* Human Product designer uses what is typically called engineering knowledge, which may consist of empirical data, rules, laws, intuition, design procedures, experience and codes of practice. There are few, if any, methods that could replace the experiential knowledge of an experienced designer. A comprehensive design environment (computational or otherwise) should provide access to these various sources of knowledge and facilitate their use.

With this background, it becomes clear that the design discipline must be closely related to information modeling. Information about design is the designer's raw material and the way that information is molded ultimately determines the design that best suits the requirements. However there are disparate information sources such as information from simulation or mechanistic models (Quantitative in nature) and experiential knowledge of an experienced designer (Qualitative in nature). To date there exist no methodology that exploits information from these different sources in a logical manner. This research is an attempt to address to create a method that can provide this integration of information.

In this research, a novel information processing paradigm is presented, that ingeniously collapses information from disparate types into a coherent self that can be used for exploring and exploiting design space. The focus of this dissertation is on using the qualitative information possessed by an expert designer in conjunction with already available quantitative data from other sources such as actual experiment, mechanistic models, and simulations to better explore the design space. The application of paradigm is demonstrated for sequential sampling process, a design space exploration technique. It is hoped that creating this paradigm will act as an enabling method to not only the design exploration process itself, but also for further research in optimization (exploitation) of design space and representation of design space (metamodeling).

CHAPTER 2

Background

While the previous chapter discusses the motivation behind the desire to explore sampling and, metamodeling techniques. This chapter is focused on the literature background of design of experiment, sampling methods, metamodeling notions and related techniques. As such, it is intended to serve a dual purpose. The first is to provide an overview of the research and scientific as well as engineering progress which has taken place to date in the area of metamodeling. A second intention is to give a survey of the state of the art in several fields that are related to sampling techniques and have helped guide the problem statements and formulation of the research questions presented in Chapter 3 of this dissertation.

2.1. Design of Experiments

In nearly every design situation, one encounters certain unknown phenomena that require physical testing to be performed concurrently with making design decisions. While engineering has produced a vast knowledge of physical principles such as stress analysis, dynamics, and heat transfer, we must often perform physical experiments to validate our design decisions. As a result, a wealth of research has been developed to empirically gather information by performing experiments. The field of study, sometimes referred to "Design of Experiments" deals with what efficient set of experiments should be run to best understand a given phenomena. **Design of Experiment (DOE)** is a structured, organized method that is used to determine a relationship between the different factors (Xs) affecting a process and the output of that process (Y). This method was first developed in the 1920s and 1930s, by **Sir Ronald A. Fisher**, the renowned

mathematician and geneticist. Design of Experiments (DOE) is also referred to as Designed Experiments or Experimental Design - all of the terms have the same meaning.

As a source of guidance for sampling methods for statistical purposes, the Design of Experiments (DOE) is a long standing and well developed field. Notable is the close DOE has tie to the linear regression and analysis of variance. It also provides benefits for other optimization and regression techniques, in that the DOE methods provide a sound basis for sampling a space efficiently [Box, et al., 1987]. Experimental design is a strategy to gather empirical knowledge, i.e. knowledge based on the analysis of experimental data and theoretical models. It can be applied whenever you intend to investigate a phenomenon in order to gain understanding or improve performance.

Design of experiment is performed in order to [Daniel Frey's reference]:

1. To estimate the performance of a system at a single point in a design space (e.g., to test whether a design satisfies a requirement on performance)
2. To seek improvement in the performance of the design
3. To create models (to build, calibrate, and/or test equations, simulations, and physical analogs)
4. To learn about a design space (e.g., to estimate the effects of a set of design parameters on a set of design performance criteria)

All though all these goals are important, this dissertation places emphasis on the last two - to create models and to learn about a design space.

Building a design means, carefully choosing a small number of experiments that are to be performed under controlled conditions. There are four interrelated steps in building a set of designed experiments:

1. Define an objective to the investigation, e.g. better understand or sort out important variables.

2. Define the variables that will be controlled during the experiment (design variables), and their levels or ranges of variation.
3. Define the variables that will be measured to describe the outcome of the experimental runs (response variables), and examine their precision.
4. Among the available standard designs, choose the one that is compatible with the objective, number of design variables and precision of measurements, and has a reasonable cost.

They can be generated automatically as soon as you have decided on the outputs, the number and nature of design variables, the nature of the responses and the number of experimental runs you can afford. Generating such a design will provide you with a list of all experiments you must perform, to gather enough information for your purposes.

2.1.1. Introduction to Some Experimental Design Terminology

Experiment An investigation that establishes a particular set of circumstances under a specified protocol to observe and evaluate the implications of the resulting observations

Experiment (*definition 2*) An investigation in which the investigator applies some treatments to experimental units and then proceeds to observe the effect of the treatments on the experimental units by measuring one or more response variables

Treatments The set of circumstances created for an experiment

Experimental Unit A physical entity or subject exposed to the treatment independently of the other units

Response Variable A characteristic of an experimental unit measured after treatment and analyzed to address the objectives of the experiment

Experimental Error Accounts for the fact that experimental units treated independently and identically will not have identical response variable measurements

Observational Study An investigation in which the investigator observes units and measures one or more response variables without imposing treatments on the individuals

Factor An explanatory variable studied in an investigation that can be set at any one of two or more values

Levels The different values of a factor

Full Factorial Treatment Design The treatments consist of all possible combinations of the levels of the factors of interest

We would like to make treatment comparisons among similar experimental units. Often there will be great variation among the experimental units available for an experiment. The effect of this variation can be controlled by grouping similar experimental units together in groups called **blocks**. Treatments are evaluated by comparing the response variable measurements for experimental units within the same block. This general design strategy is known as **blocking**.

A **completely randomized design** makes no use of blocking. For a given number of experimental units per treatment, all possible assignments of treatments to experimental units are equally likely.

A **randomized complete block design** is the simplest design that uses blocking. Treatments are randomly assigned to experimental units separately within each block in such a way that all treatments occur an equal number of times in each block.

An experiment contains **replication** if at least one treatment is applied independently to two or more experimental units.

An **observational unit** is a unit on which the response variable is measured. Observational units are often the same as experimental units, but that is not always true.

2.1.2. Random and Quasi-random Methods

When considering methods of how to sample an unknown space, random point selection comes to mind as an unbiased method that is certain to ensure the avoidance of any potential systematical errors. However, there are also some concerns and problems that come with employing such methods. First, a set of random samples may not be equally distributed, and thus the information gained about the space could vary significantly for different regions. One way around this is to employ *quasi-random* methods, which have attributes similar to random, but are based on sequels and series of selections and force a more even distribution than what a truly random method may produce.

Another aspect to consider regarding the question of whether to base regression on a random or quasi-random sample is that these methods will not necessarily sample on the boundaries of the space. It follows that the region between the most closely sampled point to the boundary, and the boundary itself, is an area where the prediction of whichever metamodel was chosen actually wanders into the realms of extrapolation. This may result in exaggerated error for the prediction within that region.

Thus, a sensible and recommended strategy for such sampling is to select the points on which to base the model via some type of DOE approach, and then to perhaps use random or quasi-random methods to generate validation cases, if this is desired.

2.2. Sampling Techniques

A sampling technique results in a specified number of data points over the design domain where the actual function is evaluated. The choice of an appropriate sampling technique is generally considered critical for the performance of any metamodeling approach. Ideally, we would prefer a sampling approach whose performance is

insensitive to the metamodeling technique; that is, the performance of which does not depend upon metamodeling technique. An important distinction is often made between sampling procedure that seek to optimize (maximize or minimize) the value of the response, and those procedures aiming to gain information about a system. A good example for the first class of methods is the sampling technique whose objective is to move sequentially to more promising regions with respect to a predefined response objective and to refine gradually the empirical model to find the optima of the response. For a second class of methods the objective of the experimentation is not to reach a decision but rather to gain knowledge about the problem. The scope of this dissertation is to tackle second class of problems. However, with some minor modifications to the methodology presented in this dissertation the methodology can be easily extended to the first class of problems too. Existing techniques for the second class of problems are presented below:

2.2.1. Full Factorial (FF)

The combination of factors and levels is often called a "sample plan." The number of samples required in full factorial design can then be calculated with a mathematical formula n^k where "n" represents the number of levels and "k" represents the number of factors. This allows all variations of the levels to occur at least once so both the main impacts and interaction impacts can be calculated.

2.2.2. Orthogonal Array (OA)

Orthogonal array sampling is a variation of full factorial sampling. It allows one to estimate the main effects of several factor and several levels in fewer number of experiment [Frey et.al, 2003]. In general, orthogonal arrays require $n(k-1)+1$ experimental trials given n factors each having k levels. They have higher precision in

effect estimation with fewer number of experiments, especially if we are concerned with the main effects of the factors.

2.2.3. Latin Hypercube Design (LHD)

A Latin hypercube sample [LHS] [Palmer and Tsui, 2001] containing n sample points (between 0 and 1) over m dimensions is a matrix of n rows and m columns. Each row corresponds to a sample point. The n_p values in each column are randomly selected— one from each of the intervals, $(0, 1/n)$, $(1/n, 2/n)$, ..., $(1-1/n, 1)$. The m columns thus obtained are then randomly permuted to yield a sample that appears random overall, but is nearly uniformly distributed in each dimension (i.e., column). Latin hypercubes are useful if one desires a sample that is predominantly random, but is uniform in each dimension separately. Latin Hypercube Design is an LHS plan which uses the midpoint of each stratum as the sample value.

2.2.4. Hammersley Sequential Sampling (HSS)

The Hammersley sequence is based on the representation of a decimal number in the inverse radix format [reference], where the radix values are chosen as the first $(m-1)$ prime numbers, m being the number of dimensions. A detailed description of this technique can be found in Kalagnanam and Diwekar [1996]. The Hammersley sequence provides a highly uniform sample containing n_p points on an m -dimensional unit hypercube. The unique feature of the HSS technique is that uniformity is preserved in all dimensions, unlike other stratified sampling techniques, such as the LHS technique described above [Palmer and Tsui, 2001].

2.3. Adaptive Sampling For Optimization

Adaptive sampling designs, also called response-adaptive designs, for statistical experiments are ones where the accruing data from experiments (i.e., the observations) are used to adjust the experiment as it is being run. Typically, decisions such as how to sample during an experiment are made and fixed in advance. However, in adaptive sampling the next sampling point is based upon results from previous sampling points. Typically, adaptive sampling is performed for the optimization purposes. Rather than establish a rigid experimental design a priori, adaptive sequential sampling techniques change the experimental design during the experimentation process. An experimental design method, such as full and partial factorial searches or latin hypercubes, may be used as the initial foundation for a sequential sampling method. Adaptive sequential sampling techniques use the results from prior experimental samples to adapt the experimental design and to select subsequent sampling locations.

Many sequential sampling techniques involve metamodels, either as an iterative result in the process or as an ultimate output of the experiment or simulation. Many approaches use one or more sequential sampling criteria and optimization algorithms to determine the next sampling point(s). The sequential sampling criteria and metamodel form are both critical decisions if an adaptive sequential sampling technique is to generate an accurate metamodel of the unknown function.

2.3.1. Sequential Sampling Criteria

Various authors have identified a variety of criteria. For instance, Kushner's criterion [Kushner, 1964] applies a Gaussian cumulative distribution function to identify the point with the greatest probability of improving upon the best point in the model.

Botev [2004] proposes an alternate approach based on an entropy model. Other authors, including Locatelli [1997], Mockus [1989] and Schonlau [1997], subsequently proposed variations of Kushner's criterion. Cox [1997] proposes an alternative criterion based on lower confidence bounding, while Watson [1995] developed a set of thresholding criteria for geological contamination studies. Many of these criteria were originally developed for geologic exploration and are associated with kriging metamodels, which are commonly employed in geostatistics. These criteria can be classified as either global or local criteria, based on their search scope. Table [2.1] summarizes the criteria reviewed by Sasena [2002b]. Sasena [2002a; 2002b] found that the most consistently and best performing technique was the switching criterion (*Switch* in Table 2.1). While for specific problems other criteria might outperform *Switch*, it consistently performed amongst the best criteria.

Adaptive sequential sampling techniques for metamodels may need to pursue multiple sampling goals simultaneously. Samples should be collected so as to provide an accurate survey of the breadth of the entire design space so that local features are not missed and to reduce the metamodel error, particularly in regions of interest. These regions of interest may include extrema such as minima or maxima, or regions that delineate critical changes in the behavior of the underlying function. For instance, regions of rapid change suggest that the behavior of the underlying function is rapidly changing and should be characterized carefully. Regions with little curvature may be of interest for

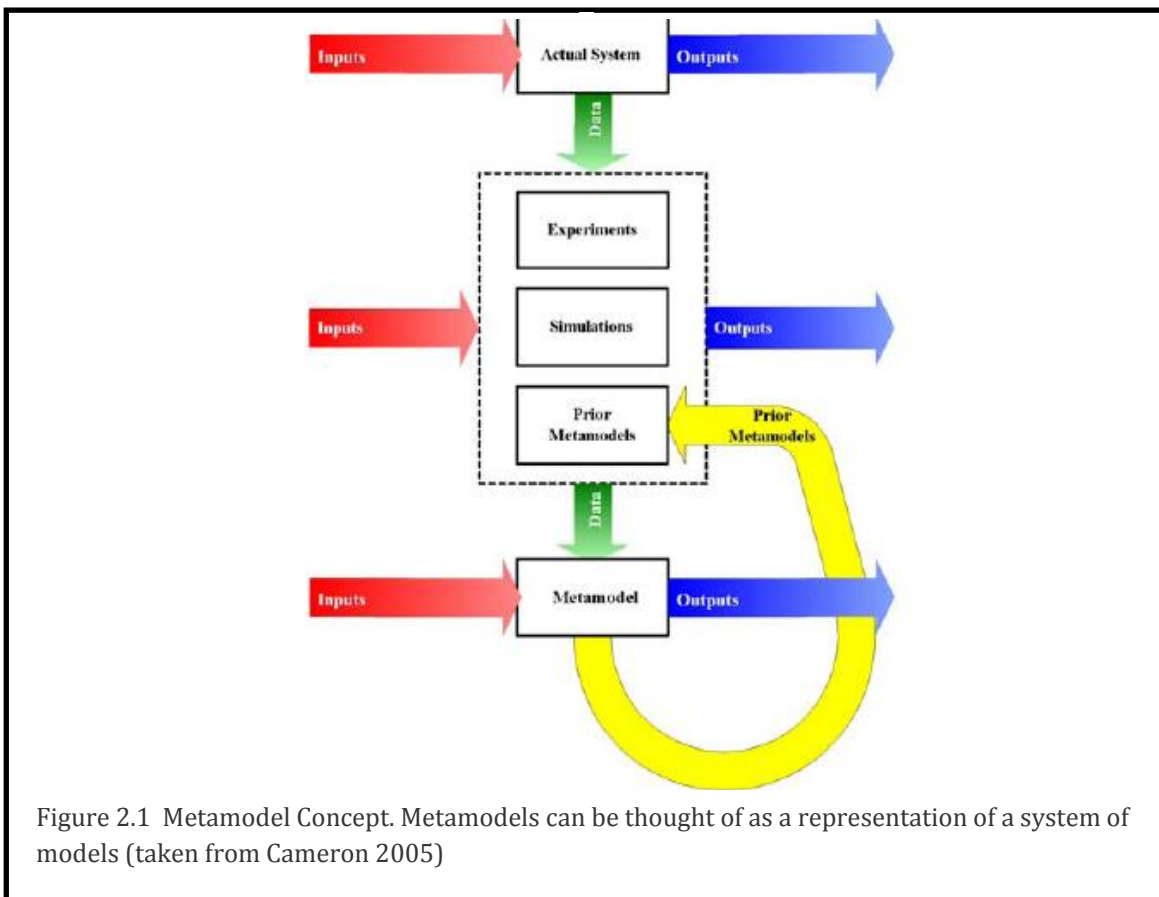
Symbol	Name	Formula	Global Model Focus	Local Model Focus
EI	Expected Improvement Function	$EI = \begin{cases} (f_{\min} - \hat{y})\Phi(z) + \hat{\sigma}\phi(z) & \text{if } \hat{\sigma} > 0 \\ 0 & \text{if } \hat{\sigma} = 0 \end{cases}$	No	Yes
EI_g	Generalized Improvement Function	$EI_g = \sigma^g \sum_{k=0}^g (-1)^k \left(\frac{g!}{k!(g-k)!} \right) z^{g-k} T_k$ and, $T_0 = \Phi(z)$ $T_1 = -\phi(z)$ $T_k = -\phi(z)z^{k-1} + (k-1)T_{k-2}$	For large g values	For small g values
Cool	EI_g with a Cooling Schedule	Same as above but g decreases with successive iterations	Initially	In final iterations
K	Kushner's Criterion	$K = \Phi\left(\frac{(f_{\min} - \varepsilon) - \hat{y}}{\hat{\sigma}}\right)$	For large ε	For small ε
lcb	Lower Confidence Bounding Function	$lcb = \hat{y} - b\hat{\sigma}$	For large b	For small b
WB_1	Watson & Barnes Threshold Bounded Criterion	$WB_1 = \Phi\left(\frac{f_{\min} - \hat{y}}{\hat{\sigma}}\right)$	No	Very local (if $\varepsilon=0$ becomes K)
WB_2	Watson & Barnes Regional Extreme Criterion	$WB_2 = \begin{cases} (f_{\min} - \hat{y})\Phi(z) + \hat{\sigma}\phi(z) + \hat{y} & \text{if } \hat{\sigma} > 0 \\ 0 & \text{if } \hat{\sigma} = 0 \end{cases}$	No	Yes
WB_3	Watson & Barnes Surprise Minimization	$WB_3 = \min_x \max_v Var[r(v) S \text{ and } x]$	Yes	No
Maxvar	Maximum Variance	$\hat{\sigma}^2(x) = \sigma^2 \left(1 - \{r_x\}^T [R]^{-1} \{r_x\} \right)$ for the Spatial Correlation Function $R(u, x) = e^{-\theta \ u-x\ ^p}$	Yes	No
Switch	Switching Criterion	1. Perform 5 iterations (with Maxvar) 2. Explore the minimum point until 3 consecutive iterations result in values within 0.1% of each other 3. Return to global iterations (step 1)	During Step 1	During Step 2

Table 2.1 Different switching criteria (taken from [Sasena 200b])

design problems, denoting robust solution regions. Other regions of the design space may play a significant role in the definition of the metamodel and should be defined carefully.

2.4. Metamodels

Metamodels, models of models, are a rapidly emerging engineering technique that offers great potential to improve engineering design. As engineering design becomes increasingly data driven, the task of identifying and uncovering fundamental relationships for a design becomes increasingly important. Metamodels are used to encapsulate information from multiple simulations or experiments, themselves models of an actual system, into a single mathematical system approximation. A metamodel can also be defined from data provided by other metamodels. Conceptually, this is shown in Figure



2.1. In essence, a metamodel is a black box representation of an unknown system (or function) [Turner, 2005]. Metamodels are desirable for use as surrogate models of actual systems. Such models are particularly attractive when the ability of an actual system to provide data is limited by the rate at which data can be obtained, when the cost of data is prohibitive or when multiple data sources must be queried for a single data point. These cases often arise in engineering design applications. For instance, in aircraft design, individual models simulating the structural loading on the airframe are combined with computational fluid dynamics simulations of airfoil performance and with wind tunnel testing of the airfoil. Each of these simulations may require hours or even days of computing time, and the cost of large numbers of wind tunnel tests often are prohibitive. In this and similar cases, it is common to adopt a metamodel to encapsulate results from a limited testing regime for design optimization purposes.

However, it would be incorrect to conclude that metamodels are only useful for scarce data applications. Metamodels are used to approximate the underlying function for nondeterministic data, or to provide approximations of underlying functional behaviors in data rich applications.

There are multiple metamodel types, including geometric models, originally derived to provide data set curve and surface representations, stochastic models that add a stochastic component to a geometric metamodel, and heuristic models that mimic cognitive processes to detect and model underlying data patterns. Each model type has advantages and disadvantages, which makes it difficult to argue that any *one* model is conclusively superior to *all* other formulations for *all* applications. Perhaps the best that can be said is that barring information about the nature of the design space to be modeled, certain metamodels are better choices than other metamodels. Since an engineer does not

necessarily have sufficient a priori insight into the behavior of a design space, identifying a metamodel likely to result in a reasonable design space approximation is important.

In subsequent sections, Response Surface Models (RSMs), Multivariate Adaptive Regression Splines (MARS), NURBs Hyperdimensional Performance Models (HyPerModels), Kriging Models (KMs), and Radial Basis Function Models (RBFs) are studied. Both interpolating and approximating forms of kriging models and RBFs are included in the literature review. Comments are also provided for several other metamodel types including Support Vector Machines (SVMs) and Neural Networks (NNs).

2.4.1. Geometric Metamodels: Response Surface Models

The simplest metamodel type is the response surface model, RSM, often first introduced as a curve fitting technique. RSMs are a natural outgrowth of Design of Experiments (DOE) techniques. A typical RSM will fit a polynomial model defined with the power series basis, as defined in Equation 2.1 as,

$$f(x) = \sum_{i=0}^n \beta_i x^i \quad (2.1)$$

where $y(x)$ is the dependent output variable modeled as an n^{th} polynomial defined by the coefficients, β_i , and the independent variable, x . A multivariate RSM form would involve additional summations, independent variables, and coefficients. Alternatives to the power series basis include trigonometric or logistic bases, which can also be used to define RSMs. An appropriate basis function must be selected a priori.

In order to fit the RSM to a data set, the degree, n , of the polynomial model needs to be defined, and the N coefficients need to be calculated. To solve for the coefficients, Equation 2.1 can be rewritten in matrix form, such that

$$y = F \beta \quad (2.2)$$

where \mathbf{y} is an $p \times 1$ vector of right hand side terms that are the output responses from the data source corresponding to the independent variable, x . \mathbf{x} is the $N \times 1$ vector of unknown polynomial coefficients, and \mathbf{F} is an $p \times N$ matrix of power series terms. These matrices are derived as part of a typical Analysis of Variance (ANOVA) procedure. Obviously, if $p = N$, \mathbf{F} is square and may be invertible, resulting in an *interpolating* polynomial. As is more often the case, \mathbf{F} is not square, but may be inverted by the use of a pseudoinverse, \mathbf{F}^+ . In this form, the resulting polynomial is a “*least squares best fit*” of the data that will *approximate* the data points rather than interpolating them.

The cost of calculating \mathbf{F}^+ , or \mathbf{F}^{-1} (if \mathbf{F} is invertible) dominates the RSM fitting cost to the data set. Fortunately, the cost of calculating the pseudoinverse is limited by the RSM order and dimensionality. Notably, the user defines the order of the RSM metamodel in advance while the number of input dimensions in the data set defines the dimensionality, d . Matrix inversion is generally considered an $O(N^3)$ operation [Press, 2002], so the key term is n , which is the characteristic dimension of \mathbf{F}^+ defined by Equation 2.3,

$$N = (n + 1)^d \quad (2.3)$$

where d is the number of independent variables, and n is the RSM degree.

The implicit RSM form is an appropriate representation for modeling unbounded curves (or higher dimensional surfaces) and exhibits reasonable predictive capabilities [Kleijnen, 2000]. RSMs are unique formulations with respect to a multiplicative constant. Membership of a data point within the RSM metamodel can be easily confirmed and the model coefficients, the β 's, contain significant information that can be useful to the analyst.

However, RSMs are rarely formulated with degrees greater than 2 (quadratic) due to excessive “*wiggling*” that may result when fitting a higher order polynomial to a data

set [Simpson, 1997; 2001; Wang, 1999a]. Consequently, their ability to accurately fit functions with locally highly nonlinear data is limited. One approach to this nonlinearity problem is to define multiple polynomial “*patches*”, with each patch valid over a local metamodel region [Nicolai, 2004]. However, the problem of maintaining differentiability and continuity between adjacent patches is non-trivial, and can result in unintended metamodel behaviors. In addition, an RSM is coordinate system dependent and consequently, different coordinate systems or system orientations, can also lead to inconsistent metamodel results.

RSMs can be used to define curves, surfaces, or hypersurfaces (surfaces in more than 3 dimensions), but cannot represent spatial curves, solids, or hyperobjects in higher dimensional spaces. While RSMs can be successfully meshed for visualization, the task is considerably more complex than for competing techniques [Piegl, 1997a]. Nonetheless, RSMs are commonly available in analysis software.

2.4.2. Geometric Metamodels: Spline-Based Metamodels

A geometric alternative to RSMs, is to replace the independent variable, x , with a parametric variable, t . Models of this type are known as spline-based models. Multiple basis functions for spline models exist, but only two types of spline basis functions are considered here. The first, cubic splines are the basis for a metamodel form known as Multivariate Adaptive Regression Splines (MARS) [Rogers, 1990] that is available as a commercial product. The second, Non-Uniform Rational B-splines or NURBs [Rogers, 1990] developed in this work as an alternative metamodeling technique.

2.4.3. Cubic Splines

Splines have their origins in the thin wood strips used by ship builders known as loftsman's splines. These strips were laid out, with the assistance of several small weights known as ducks, in a loft to represent the shape of the hull of a ship [Rogers, 1990].

When modeled mathematically, these thin beams exhibit a cubic deflection equation based on a truncated power series basis. This can be expressed with the parametric equation,

$$\mathbf{P}(t) = \sum_{i=3}^3 \beta_i t^i \text{ for } t_1 \leq t \leq t_2 \quad (2.4)$$

where $\mathbf{p}(t)$ is a vector from the origin to the point on the curve defined by the parametric variable, t . β is a vector of coefficients defining both the independent and dependent coordinates of the curve. t_1 and t_2 are called the knots of the cubic spline, which define a bounding region over which a particular set of coefficients are valid. [Rogers, 1990] A cubic spline or c-spline metamodel acts to subdivide the model space into rectangular regions, each governed by a cubic spline patch. Cubic splines are fit between data points, using four pieces of information to define the necessary 's. This information includes the locations of the two endpoints of the cubic splines and the corresponding slopes at each endpoint. By ensuring that neighboring regions share common endpoints and slopes, continuity between regions can be maintained. [Rogers, 1990]. These endpoints are *both* knots and control points, but unlike the control points in a NURBs curve, they **do not** define a convex hull property for a cubic spline curve [Saloman, 1999].

2.4.4. MARS: Multivariate Adaptive Regression Splines

Based on work performed in the late 1980s, physicist and statistician Jerome Friedman developed MARS in the early 1990s [Friedman, 1990; 1991a]. MARS has since been commercialized through Salford Systems and enjoys great popularity in the statistical analysis fields, but has not made significant inroads into engineering applications.

The key to MARS is a space subdivision scheme developed by Friedman. This scheme subdivides a space into rectangular regions. Within each region, a locally valid metamodel could be defined based on linear or cubic splines. (A linear spline truncates the power series after the linear, degree = 1, term.) Friedman chose linear and cubic splines and their truncated power series basis over other spline formulations because of the one-to-one correspondences between knots, basis functions, defined by the ‘s, and regions in the model space. [Friedman, 1991a] This one-to-one behavior complimented the subdivision scheme, allowing knots to represent the region boundaries. The current implementation of MARS by Salford Systems, predominantly uses linear splines as the basis for its metamodels [Salford Systems, 2001].

What makes MARS special is not the spline formulation per say, but the subdivision scheme developed by Friedman. This scheme allows a large model space to be subdivided into smaller regional subspaces, each with their own local metamodel. Conceptually, this makes a great deal of sense, and can reduce the global difficulties of modeling highly nonlinear of the data sets with “*reasonably*” simple metamodel bases by locally representing the function with low order approximations. Essentially, the concepts of subdivision and knots allow MARS to act as a piecewise polynomial RSM model.

MARS creates a metamodel in two phases. In the first phase, MARS subdivides the design space into a user-specified large number of regions. The boundary between each region defines a knot, which is used to define a “*hockey stick*” basis function, $BF(x)$. An inverse hockey stick basis function, $iBF(x)$, is also available. Both basis functions are defined in Equations 2.5 and 2.6.

$$BF(x) = \begin{cases} BF(x) = 0 & 0 > x - c \\ BF(x) = x - c & 0 \leq x - c \end{cases} \quad (2.5)$$

$$iBF(x) = \begin{cases} iBF(x) = 0 & 0 > c - x \\ iBF(x) = c - x & 0 \leq c - x \end{cases} \quad (2.6)$$

where c is a constant associated with the knot of each basis function.

In the second phase, MARS evaluates the contribution of each basis function to the accuracy of the metamodel, searching for the metamodel with the lowest generalized cross-validation criterion [Craven, 1979] score, defined by Equation 2.7.

$$GCV(M) = \frac{1}{N} \sum_{i=1}^N \frac{[y_i - f_M(x_i)]^2}{[1 - \frac{C(M)}{N}]^2} \quad (2.7)$$

where $GCV(M)$ is the generalized cross-validation criterion for the metamodel comprised of M basis functions, N is the number of data points to be fit to the model, y_i is the data point response at x_i , $f_M(x_i)$ is the M -basis function metamodel prediction at x_i , and $C(M)$ is the cost-complexity measure of a metamodel with M -basis functions. Generally, $C(M)=M$ is used for linear regression metamodels [Salford Systems, 2001]. Based on the results of the GCV evaluation, unnecessary basis functions are eliminated from the metamodel, resulting in a metamodel with a small number of simple expressions.

Not all metamodeling techniques require a subdivision scheme and cubic splines are not without potential problems. Careful parameterization must be performed to prevent undesirable loops and folds within the resulting metamodel and discontinuities between regions can still occur if adjacent regions do not share corners [Piegl, 1997a]. Using linear splines instead of cubic splines virtually eliminates these problems, but may dramatically increase the number of subdivisions required for equally good fit.

2.4.5. NURBs-based Metamodels

Many Computer-Aided Design/Engineering (CAD/CAE) software systems use NURBs-based representations to describe geometric objects. However, until the work by Turner [2002b], the literature shows little evidence for any similar development effort for

NURBs-based metamodels. This section will present the fundamentals of NURBs. A more complete development of the mathematics for NURBs HyPerModels is undertaken in work by Turner [2002b].

NURBs are a more generic spline formulation than the splines used in other metamodeling techniques. In fact, linear, cubic and B-splines are special forms of NURBs and can only represent irrational polynomials. Thus, they only approximate certain polynomial forms, such as conic sections.

Equation 2.8, shows a planar NURBs curve, $p(u)$, defined as:

$$\mathbf{p}(u) = \frac{\sum_{i=0}^{n_c} \mathbf{b}_i w_i N_{i,k}(u)}{\sum_{i=0}^{n_c} w_i N_{i,k}(u)} \quad \text{for } a \leq u \leq b \quad (2.8)$$

where \mathbf{b} is a vector defining the location of the i_{th} of n_c control points, w_i is a positive scalar defining the weight of the i_{th} particular control point, and $N_{i,k}(u)$ is the B-spline basis function given as a function of u . The parameter u defines a position along the curve length, which is equivalent to a point on the curve defined by the vector $\mathbf{p}(u)$. The B-spline basis function is a recursive function defined by Equations 2.9 and 2.10,

$$N_{i,k}(u) = \left(\frac{u - x_i}{x_{i+k-1} - x_i} \right) N_{i,k-1}(u) + \left(\frac{x_{i+k} - u}{x_{i+k} - x_{i+1}} \right) N_{i+1,k-1}(u) \quad (2.9)$$

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } x_i \leq u < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2.10)$$

where \mathbf{x} is the knot vector, a sequence of parameter values defining the region of control point influence within the NURBs metamodel. For the i_{th} control point, that region of influence is defined by the metamodel order, k . The B-spline basis function exhibits the behaviors defined by Equations 2.11 through 2.13.

$$\sum_{i=0}^{n+1} N_{i,k}(u) \equiv 1 \quad \forall k \text{ and } u \quad (2.11)$$

$$\text{subject to the constraint } 2 \leq k \leq n_c \quad (2.12)$$

and the definition: $0/0 \equiv 0$ (2.13)

NURBs metamodels use the control point locations, control point weights (effectively a homogeneous coordinate of the control point), knot vectors, and the curve order, k , to produce a highly flexible curve definition [Gopi, 1997].

2.4.6. Stochastic Metamodels: Kriging Models

Stochastic metamodels, including kriging and Radial Basis Functions, RBFs, approach the task of modeling the underlying function of a data set through stochastic methods. These stochastic models assume that the underlying function is actually composed of two parts, an underlying functional RSM component, and a stochastic error model such that

$$y(x) = p(x) + \varepsilon(x) \quad (2.14)$$

where $y(x)$ is the dependent variable of the metamodel, a function of the independent variable x , $p(x)$ is the underlying RSM metamodel, and $\varepsilon(x)$ is the stochastic model of the expected error in $p(x)$ with respect to $y(x)$. By combining a stochastic expression with an RSM, highly nonlinear mathematical models can be modeled.

South African geologist D.G. Krige originally developed kriging in the late 1950s and early 1960s [Isaaks, 1989]. Krige was able to use historical understanding of how ore bodies occur, in conjunction with limited sampling data from an unknown site, to predict the location and viability of an unknown ore body. Krige's original techniques were further developed during the 1970s, and by the late 1980s engineers interested in improving upon RSMs began to adapt his techniques [Sasena 1998; 2002a; 2002b].

In kriging, $\varepsilon(x)$ is defined as a stochastic process term whose uncertainty with respect to $p(x)$ is defined with a spatial correlation function (SCF) such that the covariance of two points x_1 and x_2 (in a 1D case) is

$$Cov(\varepsilon(x_1), \varepsilon(x_2)) = \sigma_z^2 R(x_1, x_2) \quad (2.15)$$

where σ_z^2 is defined as the process variance of the data that scales the SCF, $R(x_1, x_2)$, with respect to the covariance between the points x_1 and x_2 . Different choices for the functional form of the SCF can lead to different data “fits” of various qualities. Fortunately, many good choices seem to exist.

Geostatistics tends to favor SCF's similar to

$$R(x_1, x_2) = \begin{cases} C_0 + C_1 & \sqrt{(x_1 - x_2)^2} = 0 \\ C_1 e^{(-3\sqrt{(x_1 - x_2)^2}/a)} & \sqrt{(x_1 - x_2)^2} > 0 \end{cases} \quad (2.16)$$

where C_0 is referred to as the “nugget” effect (you either have a nugget or not), $C_1 + C_0$ defines the “sill” or the value at large distances from a sample point, and a defines a range of influence of the data [Isaaks, 1989]. It is these values, for a , C_0 , and C_1 , that Krige would set from historical understandings of the nature of ore bodies. Unfortunately, this type of data generally does not exist for many engineering applications.

Engineering kriging models employ SCFs such as

$$R(x_1, x_2) = e^{-\theta|x_1 - x_2|^p} \quad (2.17)$$

where θ defines the range of influence of the data ($\theta > 0$), and p defines the smoothness of the model ($0 < p < 2$) where increasing values of p lead to a smoother model. For higher dimensional models, the product of SCFs for each dimension is calculated.

With this SCF, θ , defining the range of data influence, and p , defining the model smoothness, need to be determined or set by the user. This model will interpolate the given data points without further modifications [Sasena, 2002b]. However, with minor

modifications, it is also possible to achieve a smoothed kriging metamodel that approximates noisy data, an important consideration [Sasena, 2002b].

2.4.7. Stochastic Metamodels: Radial Basis Function Models

Radial Basis Function Models (RBFs) are closely related to both kriging and spline-based models, depending on the selection of the radial basis function. Like kriging, RBFs are stochastic models, and therefore define an expression for $\epsilon(x)$ such that

$$\epsilon(x) = \sum_{i=0}^n w_i \varphi(C_i, x) \quad (2.18)$$

where w_i is the weight associated with the i th of n data points, denoted c_i evaluated with the radial basis function $\varphi(c, x)$ [Dinh, 2002]. The radial basis function $\varphi(c, x)$ may take on several forms. If it is defined as

$$\varphi(c, x) = e^{\sigma^2 \sqrt{x-c}} \quad (2.19)$$

where σ is the standard deviation of the data set composed of the points used in c , then the radial basis function model is equivalent to a DACE kriging model [Dinh, 2002]. However, other forms are also commonly used as the radial basis function, including

$$\varphi(c, x) = \|x - c\| \quad (2.20)$$

where $\| \cdot \|$ is any l_p norm such as the Euclidean 2-norm, which is particularly popular in engineering applications.

Once the user has selected an appropriate radial basis function and any underlying RSM model, Equations 2.2 and 2.18 can be combined to generate a system of equations to solve for the unknown parameters, w . However, the resulting system of equations may be singular or ill conditioned, or the matrices may be fully populated, which complicates the problem and adds to the computational cost of solving for the RBF weights [Dinh, 2002; Hussain, 2002]. It is difficult to know in advance which RBF function to choose.

As the functions used as RBFs exhibit radial symmetry, so the resulting metamodels generally do not accurately represent sharp features or discontinuities in the data [Dinh, 2002]. Like kriging, very large data sets can cause problems, and RBF models are also unsuitable for certain open (non “*watertight*”) topologies [Dinh, 2002]. Finally, since every data point affects the structure of the RBF model, like kriging, they are not local in their behavior [Dinh, 2002]. The addition of new data requires the regeneration of the entire model.

By definition, RBFs are an interpolating metamodel. However, the addition of a “*nugget-like*” parameter creates an approximating RBF metamodel. This allows Equation 2.18 to be redefined as

$$\epsilon(x) = (1 - \text{nugget}) \sum_{i=0}^n w_i \varphi(C_i, x) \quad (2.21)$$

which for a small nugget value (i.e. nugget = 0.001) defines an approximating RBF metamodel.

A recent extension of RBF metamodels is the Extended-RBF (E-RBF) metamodel developed by Mullur at Rensselaer Polytechnic Institute [Mullur, 2004; 2005]. The purpose of an E-RBF is to provide additional constraints on an RBF metamodel in order to ensure certain desirable properties (for some applications) such as convexity and unimodality (the existence of a single optima in the metamodel) for cases when such properties are known to exist within the defining data set. This behavior is achieved by using multiple basis functions, some of which may be radial and others that are non-radial in behavior, in place of the single radial basis function used in an RBF metamodel. Together, the multiple basis functions in an E-RBF metamodel act to produce a single RBF basis function with more complexity and flexibility than is available through any of the traditional RBF approaches. In their study, Mullur concluded that E-RBFs performed

as well as or better than RBFs in the cases considered [Mullur 2004; 2005]. The idea of using multiple functions to construct a metamodel in E-RBFs is closely related to the basis behind heuristic metamodels.

2.4.8. Heuristic Metamodels: Neural Network Models

Since one of the chief disadvantages to RBFs is the need to select one (or more) basis function(s), it seems reasonable to suggest that an RBF metamodel composed of all possible basis functions each with their own weight might be superior. Using machine learning approaches, each weight can be estimated and thus the appropriate basis functions selected for a metamodel. Conceptually, this might result in something like that shown in Figure 2.2. The resulting metamodel is a simple neural network.

With this architecture, it is no longer necessary to explicitly model the RSM and a stochastic component; each is simply a basis function with an associated weight. Because of this feature, neural networks are often referred to as “*universal function approximations*,” a name attributable to Vladik Kreinovich, who proved that feedforward neural networks composed of linear functions could approximate any arbitrary but smooth nonlinear function to finite precision [Kreinovich, 1991]. Construction of a neural

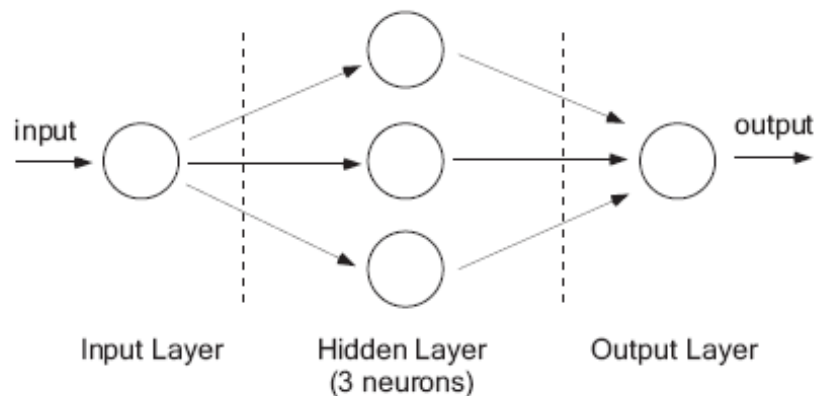


Figure 2.2 Neural Network based metamodeling

network involves three processes: selection of the basis functions to include in the network nodes, determination of the network architecture, and training the network to establish weights on each neural connection.

None of these processes is a trivial undertaking. While one might simply select a large number of basis functions to include in the network, the resulting number of nodes will make the architectural possibilities for the network expand exponentially. Even with a modest number of nodes, the architecture of the network is not necessarily as simple as shown in Figure 2.2. Several layers of nodes may be included and neurons may feed information backwards between layers. While some techniques do exist to adapt architectures to particular problems automatically [Haykin, 1999], the development of a neural network metamodel seems to remain the province of expert engineers with knowledge of neural networks.

Training is also problematic. Neural networks are derived from observations of biological learning and pattern recognition processes. Closed form solutions are difficult or impossible to obtain as the complexity of the network increases, since sufficient data may not be available. Nonlinear optimization techniques are often used to obtain a plausible (but usually nonunique) solution for the basis function weights on the basis of a portion of the available data set, and will model that portion accurately. But the data set available for training may not be adequate to fully train the metamodel [Haykin, 1999].

Defining the network basis functions, optimizing the architecture, and training the neural network can be very time consuming processes. There are no guarantees that the training set is adequate, that the nonlinear optimizations will arrive at an optimal solution, or that the solution calculated will accurately represent the desired features in the data [Inselburg, 2002]. With small data sets, a significant portion of the available data may

have to be committed to training the network, with little data remaining to validate the success (or failure) of the training.

Neural networks have a nasty habit where they identify and accurately model unintentional data patterns, and often suffer from over fitting. Over fitting occurs when a model accurately represents the data used to create it, but does not make sound predictions with respect to unmodeled data similar to the “wiggling” in RSMs. Neural network proponents often do not seem to appreciate this phenomenon and consequently do not provide adequate support for detecting over fitting issues [Myers, 2002].

While neural networks are the only metamodeling approach that offers viable solutions to problems that consist of redundant input variables, it seems that they are rarely an appropriate choice for use as a black box modeling tool. At this time, they do not appear to be suitable for the kind of black box metamodeling technique desired from this research. At the very least, proper implementation of a neural network would seem to demand the involvement of engineers with training specific to neural networks. For the interested reader, there are many excellent references on neural networks including the texts by Haykin [1999], Gurney [2002] and Mehrotra [2000].

2.4.9. Heuristic Metamodels: Support Vector Machines

Support Vector Machines, SVM, is a heuristic technique that competes quite favorably with other metamodel techniques. SVM uses a kernel function to transform the input vectors of a data set from the input space into a feature space. In the feature space, SVM attempts to identify via optimization a “flat” function that maximizes the error deviation between that function and the targets of the input data [Clarke, 2003].

The resulting function corresponds to a linear function, defined in a nonlinear space, allowing SVM to represent highly nonlinear functions with simple, linear expressions [Schölkopf, 1998]. Given a data set of N points, an SVM will generate N

coefficients corresponding to the N linear SVM expressions [Platt, 1998]. Depending upon the choice of kernel function, a number of other metamodels types can be emulated, including RSMs, RBFs and three-layer NNs [Schölkopf, 1998]. Unfortunately, SVMs still require the a priori selection of the kernel function by the user.

SVMs are expected to be particularly effective in high dimensional problems, because high input dimensionality does not result in correspondingly high dimensional kernel spaces [Drucker, 1997]. However, while SVMs have demonstrated utility in applications such as classification problems [Mangasarian, 2005], text characterization [Dumais, 1998], face detection [Osuna, 1998], pattern recognition [Burgess, 1998], principal components analysis [Schölkopf, 1998; 1999b], regression analysis [Schölkopf, 1999a; Van Gestal, 2002; Clarke, 2003], and aerodynamic data modeling [Fan, 2004; 2005], SVMs have not demonstrated the ability to significantly outperform existing techniques or to solve previously unsolvable problems [Schölkopf, 1998].

2.4.10. Heuristic Metamodels: Frequency Domain Approaches

Another interesting approach to metamodeling, particularly for dynamic systems, is to construct a metamodel using frequency domain approaches such as Fourier series or wavelets. A Fourier series, if sufficiently expanded, can represent a wide variety of functions. The problem is that the basis functions of the Fourier series are globally defined and require cancellation to define an appropriate model [Barton, 1992; 1994; 1998].

On the other hand, wavelets basis functions are locally defined. The local behavior and compact representation of wavelets makes them attractive for metamodels. However, only limited development of wavelet-based metamodels has been undertaken [Barton, 1992; 1994; 1998].

Numerous comparative studies of metamodel types have been published over the years in an attempt to determine a “*best*” model [De Veaux, 1993; Laslett, 1994; Sasena, 1998; Hussain, 2002; Turner, 2002b; Van Beers, 2004]. A few studies have compared more than three metamodel types at any one time [Barton, 1998; Wang, 1999a; Jin, 2001; Simpson, 2001; Clarke, 2003; Turner, 2005b].

In summary, the use of sampling techniques and metamodels in complex systems design is a crucial and enabling method for further research in this field. The following chapters explore state-of-the art sampling implementation process, eventually leading to a analysis of available techniques and to a formulation of a different process for design, utilizing qualitative information in conjugation with qualitative information.

Chapter 3

Research Questions and Hypothesis

This chapter introduces the research questions and scientific hypothesis. After a short review of the state-of-the-art, the research questions leading to possible improvements are identified and the requirements for an better process are formulated. Finally, the hypothesis is stated.

3.1. Design Problems Classification

What is design? One definition posited by Papalambros is that design is a decision-making process [Papalambros, 2000]. This dissertation focuses on the kinds of decisions facing the designer who must express the design space describing an artifact in order to make design decisions about some desired performance. For example, the designer could be deciding what thickness to assign a structural member that should be as light as possible while supporting the given loads.

But how does one go about describing the design space? How could one make this process more efficient? These are questions that have pushed engineers, scientists, and mathematicians to develop tools to aid designers in decision making. Many such tools belong to a field known as design optimization, whereby numerical algorithms are used to solve problems of the form

$$\begin{aligned} &\min f(x) \\ &\text{subject to: } g(x) \leq 0 \end{aligned} \tag{3.1}$$

The designer is to choose values for the vector of design variables, x , that minimize the scalar objective function, $f(x)$, while satisfying the inequality constraints, $g(x)$. Depending upon the information we have about $f(x)$, and $g(x)$, design problems can be classified into following categories:

Category 1. White Box Design Problems: If both the $f(x)$ and $g(x)$ of the design are well described in closed form equations, the design problem can be said to be a white box problem. When the functions in equation 3.1 are linear, problems of this form are referred to as linear programming problems. On the other hand if the functions in the equation 3.1 are nonlinear, problems are referred to as non linear programming problems. This category of problems are well defined and depending on nature of the problem, different optimization techniques are used to solve these problems.

Category 2. Black Box Design Problems: If both the $f(x)$ and $g(x)$ are not known in closed form, the design problem can be said to be black box problem. Since, the closed form equation of this type of problems are not known, sampling techniques and metamodeling approaches comes as aid to define and optimize this type of problems.

Category 3. Grey Box Design Problems: Generally, for any problem in hand, and especially design problems we have some qualitative information about $f(x)$ and $g(x)$ at our disposal. This qualitative information does not need to be in closed form but it should provide additional information about the problem at hand. This scenario where we do not have any closed form equation of $f(x)$ and $g(x)$ of the design problem, but have qualitative information about $f(x)$ and $g(x)$ is termed as grey box design problem.

It is conjectured that the qualitative and quantitative sequential sampling (Q2S2) technique presented here will be most beneficial in solving grey box design problems. In this dissertation, we improve sequential sampling by drawing not only on the results of

previous experiments but also on the qualitative knowledge that the user knows about the phenomenon under study.

Since information in engineering design problems can be both quantitative (QT) and qualitative (QL) in nature, combining both types of information can be more beneficial in solving real world design problems. Even though QT models are useful in providing detailed information about a design problem, they can be ineffective in situations where the mathematical formulation of design is not available or is partially defined. In such cases QL information can provide a valuable access to the design problem by taking advantages of human approximate reasoning to improve the complex design problem representation. However, all the sequential sampling approaches reported in the literature are purely quantitative in nature and thus not conducive to process information in such a mixed environment. Typical approaches to DoE involve determining the sample points all at once prior to any experimentation, or sequentially based on the QT interpretation. This dissertation proposes a mathematically rigorous methodology for handling integrated QT and QL information to suitably represent and explore search space in real world design problems.

In much of the research on sequential sampling, QT measures (such as Mean Squared Error or Maximum Likelihood Estimation) are used as criterion for selecting the next sample point[Sasena, 2002a; Jin, 2002]. These sequential sampling methods have following drawbacks:

1. Existing techniques are unable to merge information from different sources such as actual physical experiment (Non-deterministic and QT), and computer simulations (Deterministic and QT). In fact, most techniques proposed work well only for design and analysis of computer experiment (deterministic experiments).

2. Sampling strategies are dependent upon the properties of the metamodel which is used to fit the model for the data.
3. For most design problems, there exists qualitative information in the form of user guesses, first principles, monotonicity information and sensitivity information which is never exploited although readily available and useful.

These drawbacks can be attributed to the fact that there exists no method to assimilate QT and QL information coming from various sources. This necessitates the development of a method that can easily handle, merge and manipulate both QT and QL information. Secondly, the method developed should be independent or decoupled of the properties of the metamodel. In this dissertation a novel technique, which addresses the limitations of these existing techniques is proposed.

It is conjectured that our method is best when the design problem at hand has not been explored previously or the problem is highly non-linear in nature. It can also be argued that once a suitable design space is constructed it can be used to seek improvement in the performance of the design by application of suitable optimization techniques on the newly constructed design space. Furthermore, construction of appropriate model and learning about the design space can help in estimating the performance of a system at a single point in a design space.

In order to further elucidate the nature of gray box problem and role of QL information in design an illustrative example is presented next. Through this illustrative example, some basic definitions are also presented.

3.2. Illustrative Example and Definitions

Consider a projectile thrown at some elevation angle φ and initial velocity v_0 . The goal of the problem is to determine the flight distance d of the projectile in terms of these inputs. Ignoring air resistance and assuming the start and the landing position of the projectile are at the same ground level, the flight distance is given by $d = 2v_0^2 \sin \varphi \cos \varphi / g$. Qualitatively, a greater initial velocity of the projectile results in a longer flight distance. However, flight distance also depends on the elevation angle. The maximal flight distance is achieved by throwing the projectile at elevation angle $\varphi = \pi/4$ (45degrees). Qualitatively, flight distance monotonically increases with elevation angle if $\varphi < \pi/4$, and decreases with elevation angle if $\varphi > \pi/4$. In both cases, flight distance monotonically increases with initial velocity. Thus, we have two piecewise monotonicities with respect to variable φ and one full range monotonicity with respect to the variable v_0 (for φ the range considered is only between 0 and 90 degrees). It is easy to guess that with initial velocity equal to 0, the distance traversed by the projectile will be 0. Furthermore, it is also easy to guess that with elevation φ being equal to 0 or $\pi/2$ (90 degrees) i.e., if you throw a projectile directly upward or downward, the distance traversed by the projectile will be 0. As one can notice that this information about the problem is qualitative in nature and is readily available even before any blackbox or actual experimentation based approach is applied to fit a metamodel for the problem. Generally, for any problem in hand, and especially design problems we have qualitative information of this nature at our disposal. Using this data in conjunction with quantitative information available from existing sequential sampling techniques will lead to better and effective sequential sampling techniques. We are due for simple definitions.

Definition 1. *Qualitative information (QL)* is information that is descriptive in nature, relating to, or involving quality about the problem at hand.

QL information captures properties of the problem that are observed and can generally not be measured. QL description merely provides enough information to capture the distinctions that make an important difference while disregarding other information. QL information seeks to provide insights through loosely structured, mainly implicit data rather than measurements. For example, QL information does not make any assumptions regarding the linearity of the modeled system. Thus it is appropriate for the identification of non-linear systems. For the purpose of this dissertation, QL information is non-quantifiable information such as designer's expertise, designer's guesses, monotonicity information, and sensitivity information.

Definition 2. *Quantitative information (QT)* is information expressible in terms of quantity.

QT information is single numerical value or set of values that result from the output of a single point in the input space of the problem at hand. For the purpose of paper, QT information is information accrued from either actual physical experimentation or the computer simulation of the problem.

Another important facet of methodology presented in this dissertation is the incorporation of data from multiple *fidelity* sources. In order to clarify concepts related to fidelity of various sources and to avoid confusion with respect to noise generated by various sources, we present the definition of fidelity and noise.

Definition 3. *Fidelity (accuracy)* is the accuracy of the representation of the problem when compared to the real world problem.

Definition 4. *Noise (Precision)* is the difference between the predicted and observed values of a model due to random error including missing and incomplete data.

Fidelity of the model is the degree to which a model or simulation reproduces the state and behavior of a real world object or the perception of a real world object, feature, condition, or chosen standard in a measurable or perceivable manner; a measure of the realism of a model or simulation; faithfulness. Fidelity should generally be described with respect to the measures, standards or perceptions used in assessing or stating it. Fidelity can characterize the representations of a model, a simulation, the data used by a simulation (e.g., input, characteristic or parametric), or an exercise. Each of these fidelity types has different implications for the applications that employ these representations. With these definitions in place we can say that an actual physical experiment has higher fidelity than computer simulation (meta-system) even though the noise is higher in actual physical experiment.

The concepts introduced in this dissertation, can be used to merge data returned from results of varying fidelity, for example, results from detailed experiments, results from computer simulations, or results from approximations specified by the user.

3.3. Research Questions

Reevaluating the widely applied sampling process/procedure identified in Chapter II and Chapter III in this light, the following research questions arise:

1. Is there a way to combine the QL and QT information present in design problems?
2. Further, is it possible to not only integrate QL and QT information but also include fidelity aspect of information of disparate sources?

3. How can this information merging process be used for sequential sampling purposes?
4. If indeed such a process can be formulated, what are the requirements for its application?
5. Is it possible to create and formulate this implementation process in such a way that it can be employed with similar ease of implementation that other sampling techniques have impressively demonstrated in the past?

The investigation of these questions leads to the intricacies of a new sequential sampling process formulated and introduced in the subsequent sections and chapters.

3.4. Hypothesis

With regard to the previous sections, the hypothesis of this dissertation can be formulated as follows:

A sequential sampling implementation process based on merging both quantitative and qualitative information coming from varying fidelity has the potential of producing higher accuracy models for design exploration purposes.

To show the support for this hypothesis, a rigorous investigation of several application problems is in order. Here, the emphasis should be placed on:

1. The accuracy of the prediction coming from such a sequential sampling process.
2. The ability of a such a process in to solve problems in mechanical design field.

3. The relative ease of application. It is desirable that the method, once implementation conditions and parameters have been decided upon, may be employed largely as a black box, requiring that little in-depth knowledge of the method's details are actually necessary for correct application. This requirement stems from the observation that many techniques, such as Latin Hypercube Design (LHD) have not been always been widely applicable. While giving more accurate predictions as compared to other sampling techniques, the user applying these techniques needs working knowledge of LHD in order to tune the sampling process to a beneficial one. This often proves impractical, since the engineers employing these techniques in industry need to be experts within their own field to ensure proper execution of the disciplinary and system level analyses that form the basis for each approximation. Hence, the field cannot afford to require that these same people also become experts in sequential sampling techniques as well.
4. The efficient use of information. This is important because the analysis codes employed by the design engineer for system as well as disciplinary level analyses can be extremely involved and expensive, from a computational perspective.

The subsequent chapters are dedicated to investigating the background and requirements for the development of such a metamodeling process, then formulating and implementing it. This is followed by several proof-of-concept applications, including known functions which provide benchmark models, one mechanical design problem, to ensure applicability in the sense of the above outlined criteria.

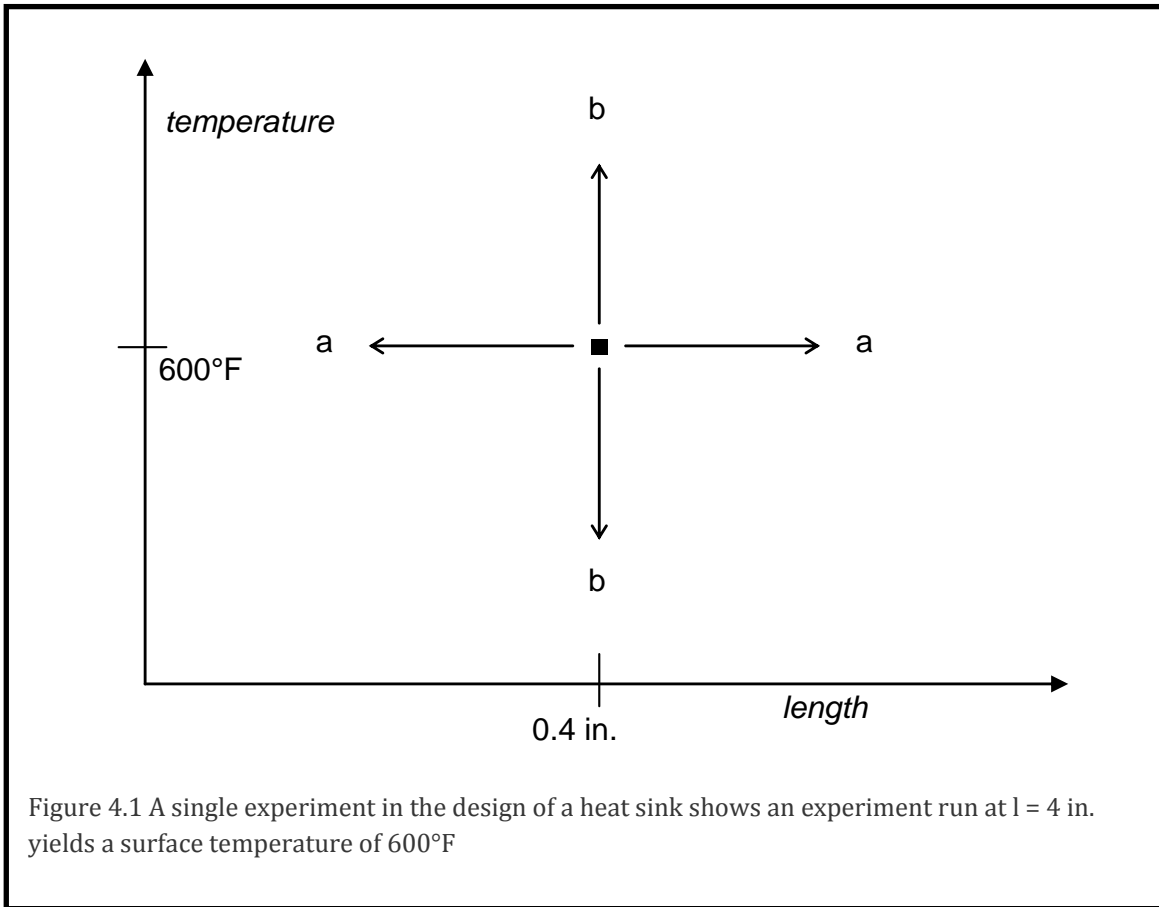
Chapter 4

Formulation of Quantitative and Qualitative Sequential Sampling (Q2S2) Process

This chapter introduces the steps needed to implement the Q2S2 technique and formulates a new implementation process to be employed in design of complex systems such as mechanical design systems. In this chapter the concept of confidence fields that captures both quantitative and qualitative data is presented. How the confidence fields assimilate data from varying fidelity sources such as actual physical experiment, computer simulation and user guesses is presented next. The next section also discusses the mathematical models and their properties pertaining to monotonicity information. Once the concept of confidence field is elaborated, Q2S2 methodology which uses the confidence field information for sequential sampling purposes is presented. Different facets of the Q2S2 methodology are elaborated and presented with a mathematical rigor where necessary.

4.1. Confidence as a Function of x and f

The basis for this method is a qualitative measure called confidence that ranges from -1 and 1. As an illustrative example, we have run a single experiment in the testing of a heat sink design (Figure 4.1) where a design variable is set to a length of 0.4 in. The result of the experiment shows a resulting surface temperature of 600°F. Given this information, one has a better sense of how other values of length will perform. Qualitatively, our confidence about values near 0.4 in. is likely to be higher than values far from 0.4 in. As a method to capture this notion, a **confidence** function is developed that starts at *one* at $l=0.4$ in. and diminishes exponentially to *zero* as we move away from



that value (in the a-a direction of Figure 4.1). Furthermore, if we consider moving vertically along the temperature direction (in the b-b direction of Figure 4.1), we would be surprised if the true value deviated significantly from the experimental value of 600°F . This is captured in the confidence function as an exponential decay to negative one. It is not that we lose confidence in the prediction but rather that we become confident that such a value is incorrect (unless, of course, the experiment was greatly flawed). For example, running a replicate experiment at $l=0.4\text{in.}$ would likely produce temperatures around 600°F , and unlikely to produce values at, say 10°F or 1000°F .

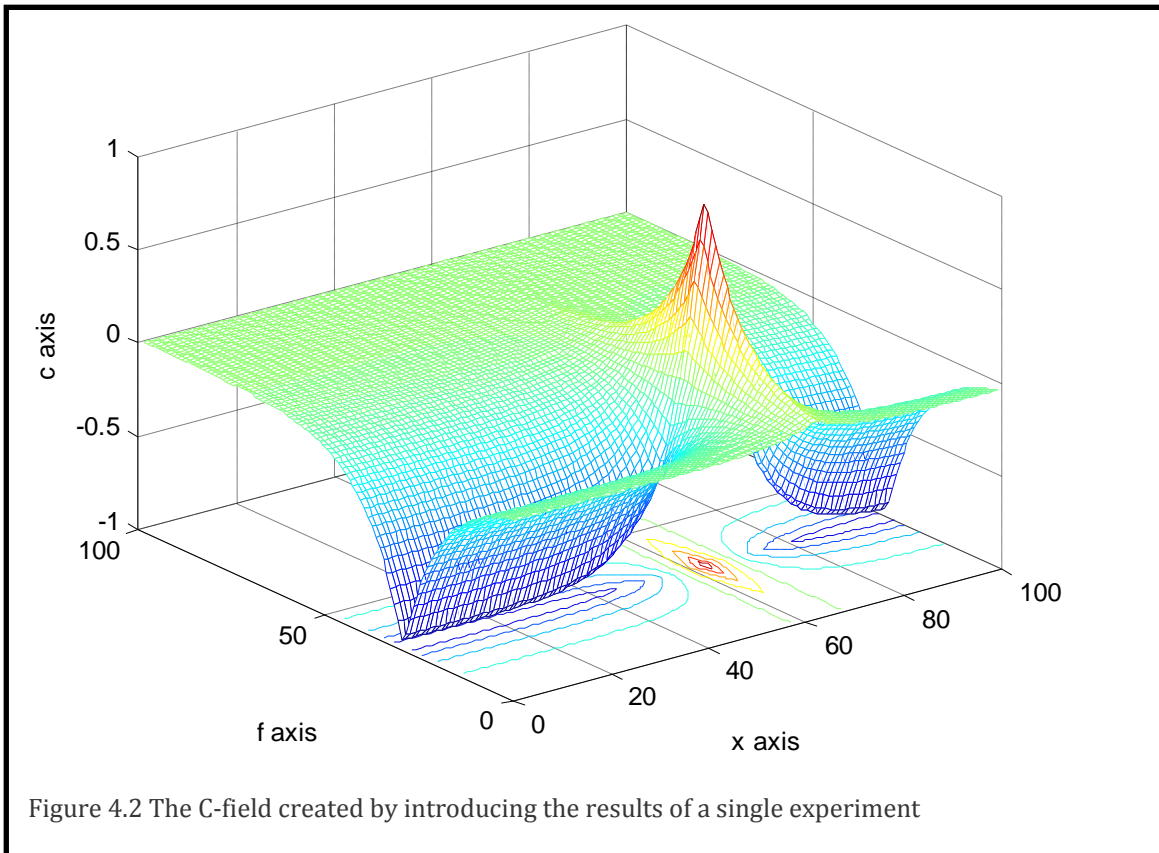
The newly defined parameter of Confidence, or C , is a function of both the decision variables or input variables, \bar{x} , and the phenomenon or output in question f . The values of C range from -1 to 1 , where:

$C[f_0 : \mathbf{x}_0] = 1$ corresponds to absolute certainty that the value of \mathbf{x}_0 does produce the value of f_0 ,

$C[f_0 : \mathbf{x}_0] = -1$ corresponds to absolute certainty that the value of \mathbf{x}_0 does not produce the value of f_0 ,

$C[f_0 : \mathbf{x}_0] = 0$ corresponds to absolute uncertainty in whether the value of \mathbf{x}_0 can produce the value of f_0 .

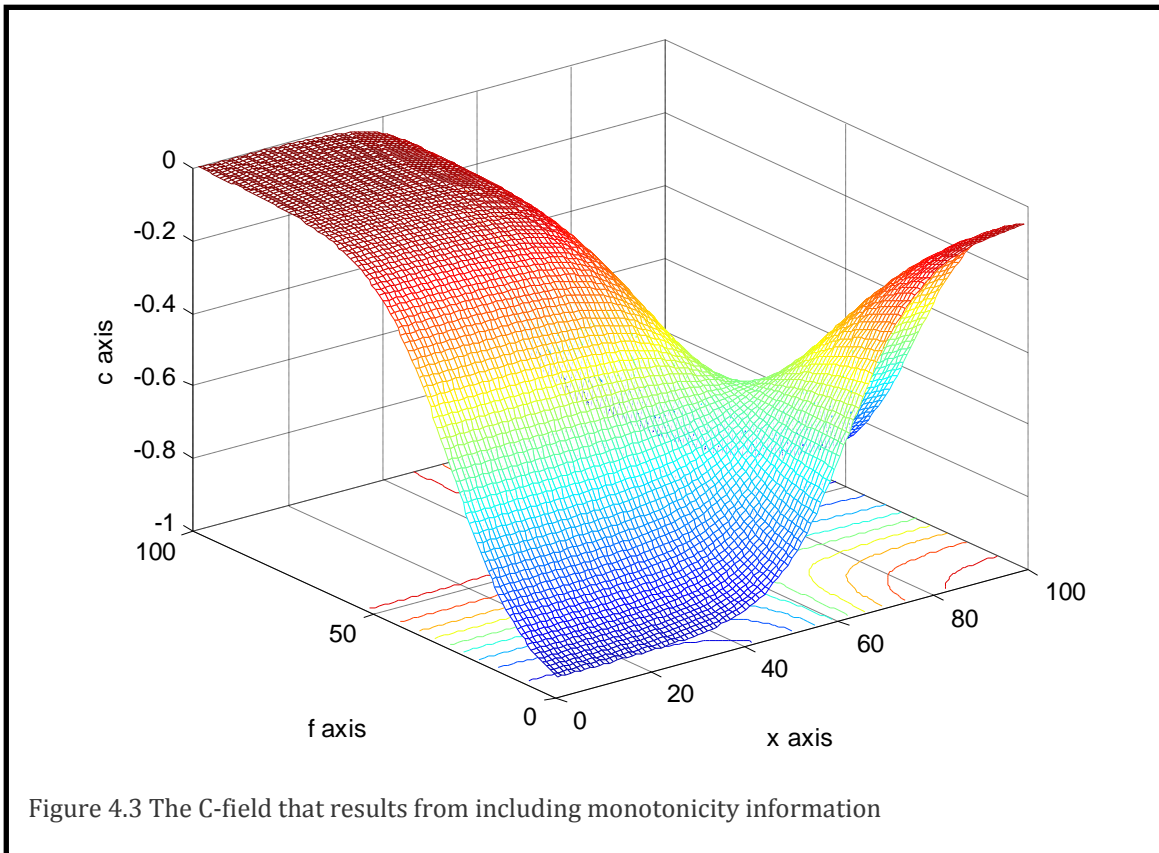
Adding this Confidence field affords more benefits than the harm introduced by



adding this new dimension. For example, one can now include information of varying degrees of fidelity. We can update the C-field to include information about detailed experiments (Figure 4.2) or monotonicity information (Figure 4.3). The equation for the C-field of actual experiment for a one variable design problem presented in Figure 4.2 is:

$$C(f : \bar{x}) = (2e^{\frac{f - f_0}{f_{tol}}^2} - 1)(e^{\frac{\bar{x} - \bar{x}_0}{\bar{x}_{inf}}^2}) \quad (4.1)$$

where, \bar{x}_{inf} or $\bar{x}_{influence}$ is similar to the σ in Kriging, and the f_{tol} or $f_{tolerance}$ is the expected error or standard deviation in the experiment due to the fidelity aspect of the information source. By varying the values of f_{tol} we can integrate the information from different sources while taking into account the fidelity issue. Typically, the f_{tol} value will



be higher for a user guess, than a computer simulation followed by an actual physical experiment in that order (i.e, $f_{tol_userguess} > f_{tol_simulation} > f_{tol_actualexperiment}$). Since, confidence is a newly defined qualitative value; other formulations may be pursued (for example, algebraic or trigonometric expressions as opposed to this exponential one). Vectors are marked with an arrow and are used to represent an N -tuple, i.e., $\vec{u} = u_1, u_2, \dots, u_N$.

The concatenation of two vectors \vec{u} and \vec{v} of size N and M , respectively, yields a vector of size $N+M$ denoted by $\vec{u}:\vec{v} = u_1, u_2, \dots, u_N, v_1, v_2, \dots, v_n$. C is a function of both the decision variables, \vec{x} (I tuple), and the phenomenon in question, f (1 tuple for one objective function). For actual experiments, simulation or user guesses the equation for the C-field as a function of both \vec{x} and f and can be written in the convolution form as follows:

$$C_e(f : \vec{x}) = f_f(u_f) \cdot x_1 u_1 \cdot x_2 u_2 \cdot \dots x_N(u_N) \quad (4.2)$$

$$\text{Or } C_e(f : \vec{x}) = f_f(u_f) * \prod_{i=1}^I x_i u_i \quad (4.3)$$

where the subscript e denotes that the c-field is for actual experiment or user guess. The function can be written as follows:

$$f_f(u_f) = (2R_f(d_f) - 1) \quad (4.4)$$

The subscript f here denotes the dependence of C field on the objective function, and $x_i u_i = R_i(d_i)$.

$$\text{Where, } d_f = \frac{f - f_0}{f_{tol}} \text{ and } d_i = \frac{x_i - f_{i0}}{x_{itol}} \quad (4.5)$$

More specifically, there could be 8 different functional choices for $R(d)$ as presented in Table 4.1. For one objective function and 3 variables if $R(d)$ is same for all

Name	$R(d)$
Exponential	$\exp(- d)$
General Exponential	$\exp(- d ^{n+1}), 0 \leq n \leq 2$
Gauss	$\exp(-d^2)$
Linear	$\max(0, 1 - d)$
Spherical	$1 - 1.5 d + 0.5 d ^3$ for $ d \leq 1$, 0 otherwise
Cubic	$1 - 3 d ^2 + 2 d ^3$ for $ d \leq 1$, 0 otherwise
Spline	$1 - 15 d ^2 + 30 d ^3$ for $0 \leq d \leq 0.2$ $1.25(1 - d)^3$ for $0.2 \leq d \leq 1$ 0 for $ d > 1$
where $ d $	

Table 4.1 Different functions for representing C fields.

and is of Gaussian form then the C-field equation for a given experiment or user guess can be written as follows;

$$C(f; \vec{x}) = (2 * \exp(-d_f^2) - 1) * \prod_{i=1}^3 \exp(-d_i^2) \quad (4.6)$$

where d_f and d_i are given by equation 4.5.

The actual C function could be formulated by using any of the 8 mentioned form of d for f and any \vec{x}_i . Different forms of $R(d)$ can be used in the same problem as well. For example, for 2 variables and 1 objective function, the 1st variable can have a Gaussian form whereas the second variable can have spline form for d . The f can also have any of the eight forms.

The f part in equation 4.2 varies from -1 to 1 (Figure 4.2) and the \vec{x} part varies from 0 to 1 (Figure 4.2). Note that in all cases the function decreases with increasing $|d|$.

The functions in Table 1 can be separated into two groups, one containing functions that have a parabolic behavior near the origin (gauss, cubic, and spline), and the other containing functions with a linear behavior near the origin (exponential, linear, and spherical). The parabolic functions are continuous and hence differentiable, whereas the linear functions are discontinuous. The choice of function should be motivated by the underlying phenomenon, e.g., a function we want to optimize or a physical process we want to model. If the underlying phenomenon is continuously differentiable, the function will likely show a parabolic behavior near the origin, which means that the Gaussian, the cubic, or the spline function should be chosen. Conversely, physical phenomena usually show a linear behavior near the origin, and exponential, general exponential, linear or spherical would usually perform better.

In Figure 4.3, a plot is shown for monotonicity information. The equation for the C-field shown in the Figure 4.3 is inspired by fuzzy-set theory [Zimmerman, 2001], where a 1-dimensional sigmoid function is used to black out regions that are impossible to reach. This monotonicity information can be provided through user queries. For example in the heat sink problem, a user may claim that the temperature decreases with longer lengths. Given a single experiment at length equal to 0.4 in. ($\mathbf{x}_0 = .4$) and temperature at 600⁰C ($f = 600$), the C-field can be adjusted at that point to eliminate the areas that are impossible due to this monotonicity. The equation for the C-field for one variable monotonicity information is shown in Figure 4.2b and is represented by the following equation:

$$C(f : \bar{x}) = \frac{1}{1 + e^{\left| \frac{f - f_0}{f_{tol}} \right|}} \frac{1}{1 + e^{\left| \frac{\bar{x} - \bar{x}_0}{\bar{x}_{inf}} \right|}} \frac{1}{1 + e^{\left| \frac{f - f_0}{f_{tol}} \right|}} \frac{1}{1 + e^{\left| \frac{\bar{x} - \bar{x}_0}{\bar{x}_{inf}} \right|}} \quad (4.6)$$

We consider two types of monotonicity information i.e., positive monotonicity and negative monotonicity. The C-field value for monotonicity information ranges from 0 to -1. We use sigmoid and erf function from the mathematical literature to represent the monotonicity information. For the convenience purpose let us define following:

$$sigmoid(y) = \frac{1}{1 + e^{-y}} \quad (4.8)$$

$$sigmoid(-y) = \frac{1}{1 + e^y} \quad (4.9)$$

$$erf(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{-t^2} dt \quad (4.10)$$

$$sigmoid(y) = ((erf(y)/\sqrt{2}) + 0.5) \quad (4.11)$$

$$sigmoid(-y) = 1 - sigmoid(y) \quad (4.12)$$

Then equation for representing C-fields of negative monotonicity in terms of sigmoid function can be written as follows:

$$C_{mon}(f : \bar{x}) = \prod_{i=1}^I (d_f)^* (d_i) \prod_{i=1}^I (d_f)^* (d_i) \quad (4.13)$$

And the equation for representing C-fields of positive monotonicity in terms of sigmoid function can be written as follows:

$$C_{mon}(f : \bar{x}) = \prod_{i=1}^I (d_f)^* (d_i) \prod_{i=1}^I (d_f)^* (d_i) \quad (4.14)$$

Where d_f and d_i are defined by equation 4.5.

Then the equation for representing C-fields of negative monotonicity in terms of erf function can be written as follows:

$$C_{mon}(f : \bar{x}) = \prod_{i=1}^I (d_f)^* (d_i) \prod_{i=1}^I (d_f)^* (d_i) \quad (4.15)$$

And the equation for representing C-fields of positive monotonicity in terms of erf function can be written as follows:

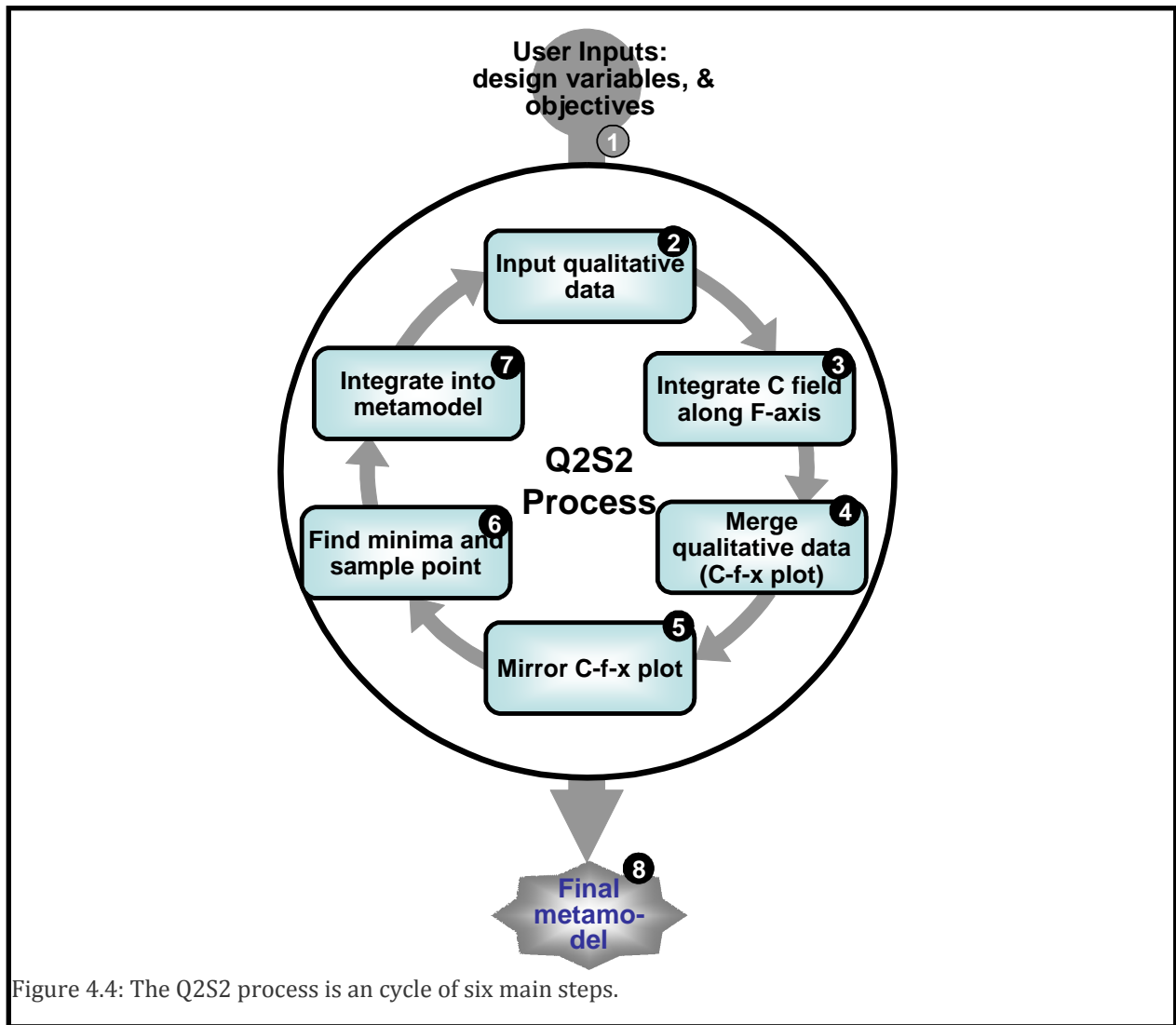
$$C_{mon}(f : \vec{x}) = \prod_{i=1}^I (d_f)^* (d_i) \prod_{i=1}^I (d_f)^* (d_i) \quad (4.16)$$

The (y) and (y) part varies from 0 to 1 (Figure 4.3). The (y) and (y) part varies from 1 to 0 (Figure 4.3). Note that the sigmoid has a range of $[0, 1]$ and erf function have a range of $[-1, 1]$. Appropriate transformation has been made to both sigmoid and erf function so that they exhibit the properties mentioned(above). The actual C function for both positive and negative monotonicity could be formulated by carefully mixing (y) , (y) , (y) and (y) for f and any \vec{x}_i . For example, for a 2-variables \vec{x} and 1 objective function, the 1st variable can have a (y) form with respect to f whereas the second variable can have (y) form. The C-field values of monotonicity function always have a range $[-1, 0]$. This is because the monotonicity function just gives us the appropriate information where the solution can not lie (or the space that need not be searched) but it does not provide any information about the space where the solution lies.

In the next section Q2S2 methodology which uses the models of information presented in this section for sequential sampling purposes is presented. Different facets of the methodology are elaborated and presented with a mathematical rigor where necessary.

4.2. Qualitative and Quantitative Sequential Sampling (Q2S2)

The methodology begins by taking inputs from the user about number of variables and objective functions (Step 1 Figure 4.4). Then it interacts with user by querying for specific qualitative information (Step 2 Figure 4.4). Our current implementation includes the following interactions with the user:



Interval Information: The program begins by asking about the ranges of \bar{x} and f it needs to consider. Once the user provides the ranges of \bar{x} and f (or the ranges they think are relevant to the problem at hand) the program proceeds to ask further questions.

Monotonicity Information: Once the interval information is inputted, the program asks the user whether he/she has monotonicity information. If the user has any monotonicity information the program represents this information as a C-field described

in equation 4.6. The monotonicity information handled by the program can be for any number of decision variables. For example, if a user knows that the problem at hand is monotonic with respect to only one x in a three variable problem but not with respect to the other two variables, the program can still capture and use this information. The process asks user to explicitly input the type of monotonicity (positive or negative). The code written can also capture piecewise monotonicity (if any) for each x .

Actual experiment or simulation data: The program also asks about the values of \bar{x} and f for which an actual experiment or simulation was performed. The distinction between the simulation and actual experiment is captured by the f_{tol} values of equation 4.1. The tolerance value for computational simulations may be kept relatively high, whereas the tolerance value for actual experiment is kept small to account for experimental variations. The program asks for the tolerance values if the data is from actual experiment.

Approximation or user guess: The program then proceeds to ask user whether they know of any approximate values at the current stage. Any approximations, no matter how imprecise can improve the model in location where no information currently exists.

It is important to note that experiments exist only at values of \bar{x} . and not at values of both x and f . Therefore, in Step 3 (of Figure 4.4), the methodology integrates C-field along the f dimension for each \bar{x} to produce a new value of confidence as a simpler function of just \bar{x} :

$$C(\bar{x}) = \int_{f_{\min}}^{f_{\max}} C(f : \bar{x}_0) df . \quad (4.17)$$

Then the methodology progresses to step 4 (details in Subsection 4.2.1), where it merges all the information at hand to create a single overall C' - f - \bar{x} function for the problem. In Step 5, the methodology mirrors the value of the constructed C' - f - \bar{x} function.

Since C' ranges from -1 to 1, the mirroring function reflects the negative C' values to positive values, thus making it now range from 0 to 1. This is done to identify where values are closest to zero since zero corresponds with no knowledge about the model. This mirroring, represents one of the main advantage in creating confidence as a field ranging from -1 to 1. Not only can this range represent whether a relationship exists or not, but it can also be normalized to identify where the least (and most) information is known (details in Subsection 4.2.2). Based on this new refined C , Step 6 performs an optimization using simple gradient based search techniques to identify the minimizer of C as the subsequent sample point. This sampling point is then integrated into the previous metamodel (which may be Kriging or B-splines). The Steps 2-8 are repeated till the number of sampling points we can expend is exhausted or when there is no more information available. The output of the process is both the resulting C -field and the associated metamodel.

The Q2S2 was implemented using MATLAB[®] R12 (Details in Appendix A). The optimization toolbox in MATLAB[®] was used for the search process. In next subsections we discuss mathematical details pertaining to various steps of Q2S2 methodology. The next section presents example problems. The optimization space appears to become multimodal relatively quickly thus complicating the search for the global optimum. It is important to note that the minimization of the co-variance in Kriging requires a similar optimization process, and to solve the values of co-variance for each point sampled requires the set up and solving of an n -by- n matrix (where n is the number of existing sample points). Similarly in our approach, the combined C -field expression is indeed composed of n terms; however, by using the exponential equation we can create an analytical expression of not only our C -field but of its gradient and Hessian as well. In doing so, gradient based optimization methods are used quite efficiently since there is no

need to find such information with finite difference approaches. The multimodality of the space is alleviated by initiating parallel optimization at the known local maxima. These maxima are known since they correspond to the peaks caused by the actual experiments.

4.2.1. Merging Functions

Step 4 of the flowchart (Figure 3) requires that different C-field information gathered from different sources or during different iterations to be merged to create a unified C-field. However, unification of C-fields should have certain properties to be meaningful. These properties are mentioned below:

Communicative and Associative property: When a unified C-field depends on *set* of information gathered but not on the *order* in which they are gathered, the function that realizes the combination, must be *commutative* and *associative*. A function which has $f(a) \ f(b) \ f(b) \ f(a)$ is said to be *communicative*. The function which satisfies $f(f(a,b),c) \ f(a, f(b,c))$ relation is said to be *associative*.

Monotonic: From the definition of C-fields, greater value of a C-field at a point from a given source should lead to higher unified C-field values. This simply says that the combination function f should be *monotonic* with respect to two given arguments: if $a > b$, then $f(c, a) > f(c, b)$ for all c .

Continuity: Smoothness of this function is essential, for if the C-field region is an interval of reals, there is no reason that one point should have drastically different effect to the combination than its close neighbors. Thus, *continuity* is also required for f .

Differentiability: Differentiability may play a role in selecting smoother functions among continuous ones. It is useful property for analytical

derivation of gradient information which can be used by gradient based optimization techniques.

Range: The combination function should be such that it does not violate the definition of C-field, i.e., for two given point $a, b \in [-1, 1]$ the combination function $f(a, b) \in [-1, 1]$.

Three different merging functions that exhibit the property mentioned above can be used to merge different C-fields:

Odd P-Normalizing merging function:

$$f(a, b) = (a^p + b^p)^{1/p} \quad (4.18)$$

Where, p is odd integer. The restriction of p being an odd number is due to fact that range is $[-1, 1]$. Merging a negative number with positive number can be preserved only when p is odd.

Averaging function:

$$f(a_1, a_2, \dots, a_n) = \frac{f(a_1) + f(a_2) + \dots + f(a_n)}{N} \quad (4.19)$$

Where, N is total number of C-fields data.

Hamacher Family:

For this class of functions,

$$g(a) = \log\left(1 + \frac{a}{1-a}\right) \quad (4.20)$$

Where, a is any constant in $(0, 1)$ and range of a is $(0, 1)$

$$\text{Then the function, } f(a, b) = g^{-1}(g(a) + g(b)) \quad (4.21)$$

have all the properties mentioned above. Equation 4.21 can be re written in the following form:

$$f(a, b) = \frac{a + b + (1 - 2)ab}{1 - (1 - 1)ab} \quad (4.22)$$

However, this function has all the properties only for the range (0, 1). In order for hamacher functions to be valid merging functions in the range (-1, 1), the equation 21 can be modified and used as follows:

$$f(a,b) = \frac{a+b-(2)ab}{1-(1)ab} \quad a,b \in [0,1] \quad (4.23)$$

The fact that equation 4.23 must be defined in segments shows that the g function is not smooth. Indeed the derivative of g is $g'(a) = \frac{1-(2)|a|-(1)a^2}{1-(1)a^2}$ (4.24)

Only when $\alpha=2$, this derivative is differentiable everywhere. This gives the only member in the Hamacher family that is smooth enough to be represented in a single algebraic formula. This is indeed the one used as the velocity combination formulae in relativistic physics:

$$f(a,b) = \frac{a+b}{1+ab} \quad (4.25)$$

For analytical brevity, ease of differentiation and due to savings in computational efforts odd P-Normalizing function was selected to proceed further.

4.2.2. Mirroring Function of C-f-x plots

Step 5 of the flowchart (Figure 4.4) requires different mirroring functions. There are two mirroring function that can be used.

Absolute Function: $|C(f : \bar{x})|$ (4.26)

Square Function: $C(f : \bar{x})^2$ (4.27)

Out of these two, square function is preferred over the absolute function due to following reasons:

Using Absolute function has the pitfall of creating potential cusps in the overall C'-f-x plot.

The Square function is more amenable for calculating the derivate of analytical expressions.

4.3. Analytical Expression Derivation

For the purpose of analytical expression derivation, the Gaussian form was selected for actual experiment/user guess information. The sigmoid function was selected for representing both the positive and the negative monotonicity information. The primary reason behind these selections was the ease of differentiability of these functions. These functions can be easily differentiated to provide Gradient and Hessian information which can be used by the gradient based optimization technique. In case we want to use algorithms which do not require gradient information any of the above mentioned function can be used for actual experiment, simulation and user guess(Table 4.1) and erf function could be used for monotonicity information purposes.

Let us assume the actual experiment or user guess information is modeled through Guassian form in Table 4.1. Then for I variable problem the actual experiment/ user guess information can be modeled by.

$$C_e(f : \bar{x}) = (2e^{d_f^2} - 1) * \prod_{i=1}^I e^{d_i^2}, \quad (4.28)$$

Where d_f is given by equation 4.5.

$$\text{Let } \prod_{i=1}^I e^{d_i^2} = A, \text{ Then, } C_e(f : \bar{x}) = A(2e^{d_f^2} - 1) \quad (4.29)$$

Let us assume the negative monotonicity information is modeled through sigmoid form in equation 4.13. Then for I variable problem the negative monotonicity information can be modeled by

$$C_{mon}(f : \bar{x}) = \prod_{i=1}^I (1 - e^{-d_i})^{-1} \prod_{i=1}^I (1 - e^{-d_i})^{-1} \prod_{i=1}^I (1 - e^{-d_i})^{-1} \prod_{i=1}^I (1 - e^{-d_i})^{-1} \quad (4.30)$$

$$B(1 - e^{-d_f})^{-1} C(1 - e^{-d_f})^{-1}$$

Where $B = \prod_{i=1}^I (1 - e^{-d_i})^{-1}$ and $C = \prod_{i=1}^I (1 - e^{-d_i})^{-1}$

Let us assume the positive monotonicity information is modeled through sigmoid form in equation 4.14. Then for I variable problem the positive monotonicity information can be modeled by

$$C_{mon}(f : \bar{x}) = \prod_{i=1}^I (1 - e^{-d_i})^{-1} \prod_{i=1}^I (1 - e^{-d_i})^{-1} \prod_{i=1}^I (1 - e^{-d_i})^{-1} \prod_{i=1}^I (1 - e^{-d_i})^{-1} \quad (4.31)$$

$$B(1 - e^{-d_f})^{-1} C(1 - e^{-d_f})^{-1}$$

Where $B = \prod_{i=1}^I (1 - e^{-d_i})^{-1}$ and $C = \prod_{i=1}^I (1 - e^{-d_i})^{-1}$

In step 3 (equation 4.17), the process integrates C-field along the f dimension for each x to produce a new value of confidence as a simpler function of just x :

$$C(\bar{x}) = \int_{f_{min}}^{f_{max}} C(f, \bar{x}_0) df.$$

Which can be rewritten in terms of d_f by simple substitution

$$C(\bar{x}) = \int_{f_{min}}^{f_{max}} C(f, \bar{x}_0) dd_f \quad (4.32)$$

This integration when performed on experiment/ user guess information gives

$$C_e(\bar{x}) = C_e(f : \bar{x}) dd_f = A(2e^{-d_f^2} - 1) dd_f = A(\sqrt{e} \operatorname{erf}(d_f) - d_f) C \quad (4.33)$$

Here A is constant with respect to d_f as it is just a function of the input variable x 's and not f . Similarly, this integration when performed on negative monotonicity and positive monotonicity equations we have,

$$C_{mon}(\vec{x}) = C_{mon}(f : \vec{x}) dd_f = B \ln \frac{e^{d_f}}{1 - e^{d_f}} - C \ln \frac{e^{d_f}}{1 - e^{d_f}} - C \quad (4.34)$$

$$C_{mon}(\vec{x}) = C_{mon}(f : \vec{x}) dd_f = B \ln \frac{e^{d_f}}{1 - e^{d_f}} - C \ln \frac{e^{d_f}}{1 - e^{d_f}} - C \quad (4.35)$$

Let us assume that when we reach step 4, we have L user guesses or actual experiment data, M positive monotonicity data, and N negative monotonicity data, and assuming normalized merging function being used for merging purpose, then mathematically step 4 can be written as follows:

$$C'_{merge}(\vec{x}) = \left(\prod_{l=1}^L a_l \prod_{i=1}^I A_{li} \prod_{m=1}^M (1)^p b_m \prod_{i=1}^I B_{mi} \prod_{i=1}^I c_m \prod_{i=1}^I C_{mi} \prod_{n=1}^N (1)^p b_n \prod_{i=1}^I B_{mi} \prod_{i=1}^I c_n \prod_{i=1}^I C_{mi} \right)^{1/p} \quad (4.36)$$

Where, different coefficient a_l , b_m , c_m , b_n , and c_n are found from evaluation of integrations in step 3. The final equation at step 5 is just the square (assuming square function being used for mirroring) of the previous equation. The equation used during the optimization step can be written as:

$$C'_{merge}(\vec{x}) = \left(\prod_{l=1}^L a_l \prod_{i=1}^I A_{li} \prod_{m=1}^M (1)^p b_m \prod_{i=1}^I B_{mi} \prod_{i=1}^I c_m \prod_{i=1}^I C_{mi} \prod_{n=1}^N (1)^p b_n \prod_{i=1}^I B_{mi} \prod_{i=1}^I c_n \prod_{i=1}^I C_{mi} \right)^{2/p} \quad (4.37)$$

This equation in analytical form can still be differentiated to provide slope and Hessian information which can be easily used by gradient based search techniques.

To illustrate the efficacy of the proposed Q2S2 methodology, chapter 5 presents results on various numerical problems, and chapter 6 presents a bi-stable MEMS switch design problem. Through these examples various aspect of Q2S2 are outlined.

Chapter 5

Implementation Example for Known Functions

This chapter gives a step-by-step overview of the Q2S2 method as well as single and multi-variable function examples. Further, it provides insight into the Q2S2 mechanism itself, and thus describes various facets of Q2S2 process using simple example functions. Another purpose of this chapter is to establish a basis of comparison, based on which we can compare different sampling techniques.

5.1. The Single-Variable Functions

5.1.1. Example Function 1

The first function modeled is a variable function and is obtained from Sasena [2002a] and Turner [2005], so that all trends can easily be visualized. Its equation is:

$$f(x) = 10 \sin(x) e^{x/100} \quad (5.1)$$

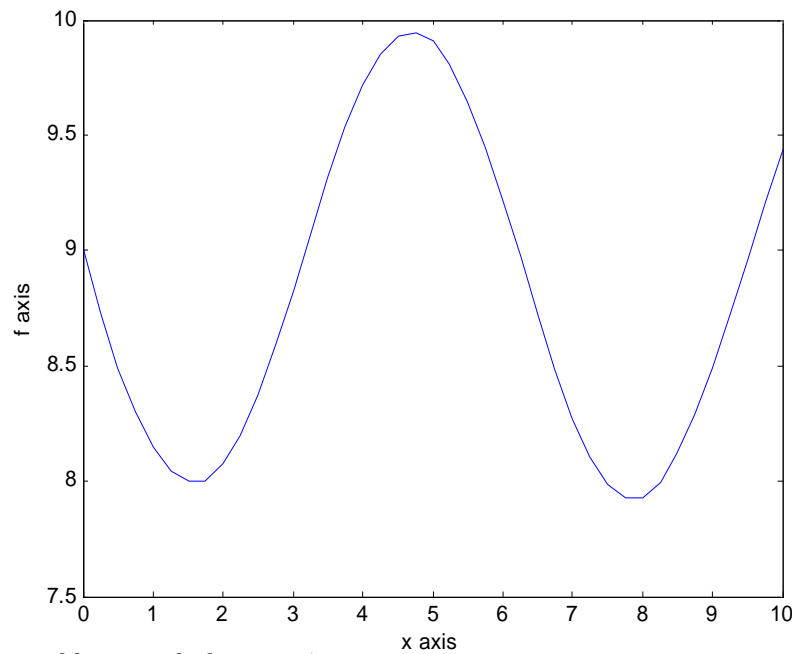


Figure 5.1 One variable example function 1

This function is an excellent test of the metamodels, with two distinct, but very similar local optima. Accurately modeling these optima is very important if the metamodel is to be later used for optimization (Figure 5.1) . For this example function, it is assumed that there is no qualitative information available and hence the problem is purely a black box for an unknown process. This is done to test the ability of the Q2S2 in the absence of any user information. This example function is also used to illustrate the working of Q2S2 methodology.

Initially two sample points are selected at minimum of x and maximum of x to proceed with. At this point the metamodel and the actual function is shown in Figure 5.2. The C-field created for these sample points by Q2S2 methodology is presented in Figure 5.3 and 5.4 respectively.

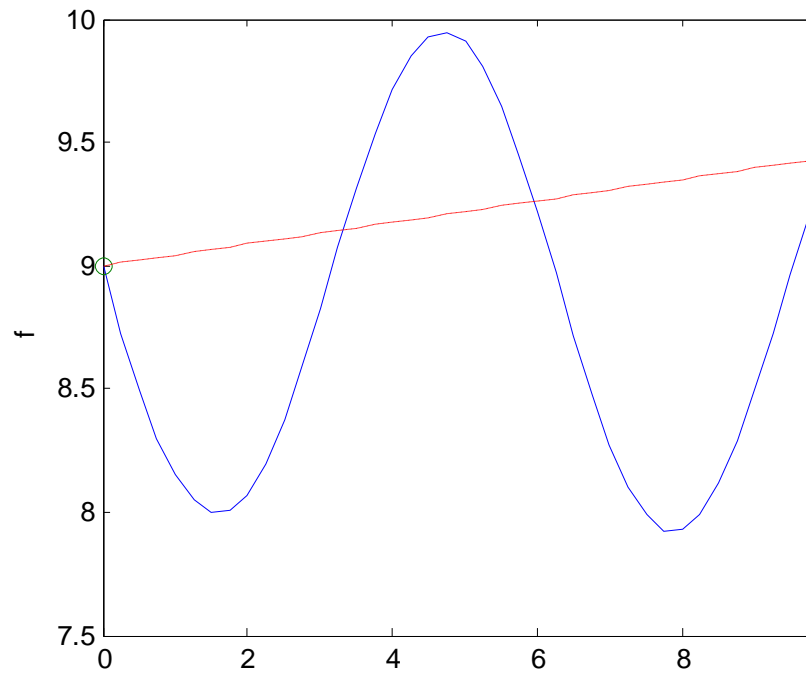


Figure 5.2 Metamodel after 2 sample points

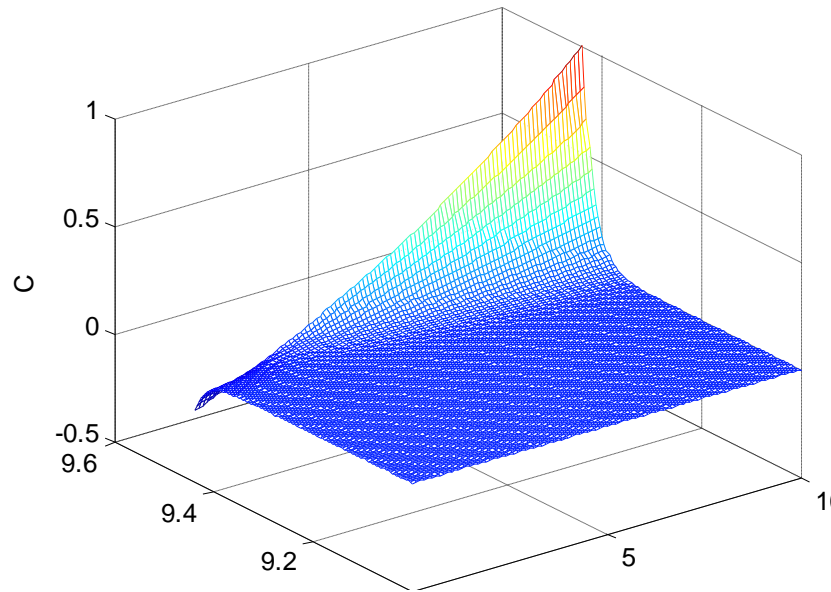


Figure 5.3 C field for the first sample point for example function 1

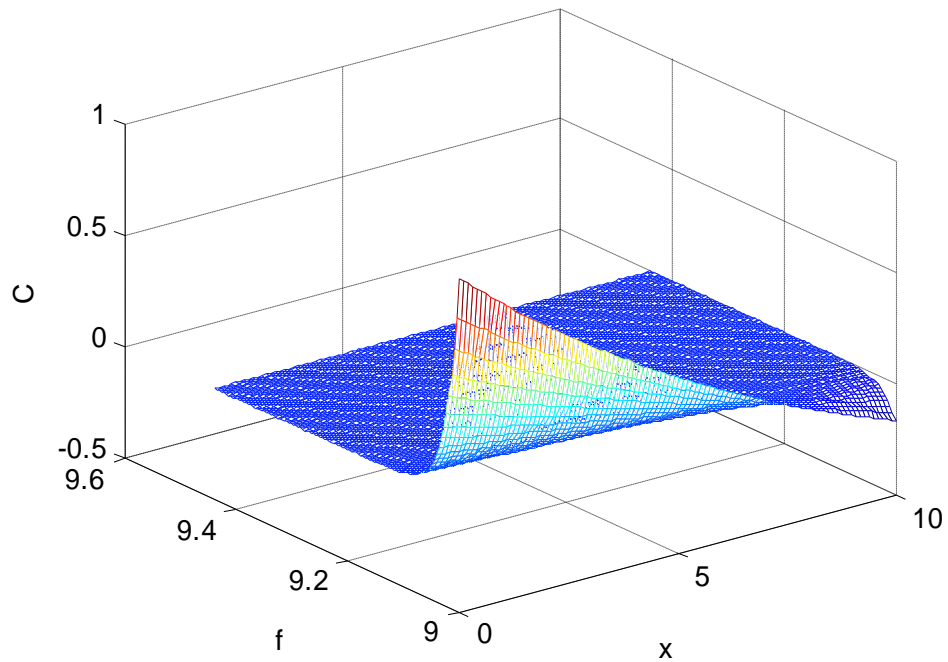


Figure 5.4 C field for the second sample point for example function 1

At step 3 of Q2S2 methodology the C' values for these sample points are calculated

(presented in figure 5.5 and 5.6). The overall C' field after the application of merging and mirroring function at this point (Step 4 and 5 of Q2S2 (Figure 4.4)) is presented in Figure 5.7. A simple optimization process (fmincon or UMDIRECT function in Matlab) is then used to find the minima of the C' field. The optimization process is started in parallel at the two peak points in the Figure 5.7. The best value of the minima coming from all the parallel starting points is selected. The location of this minima is illustrated in Figure 5.7. This point is then selected as the next sampling point. A black box evaluation of the function is carried out at this sample point. This newly sample point is then integrated into overall new metamodel (Figure 5.8).

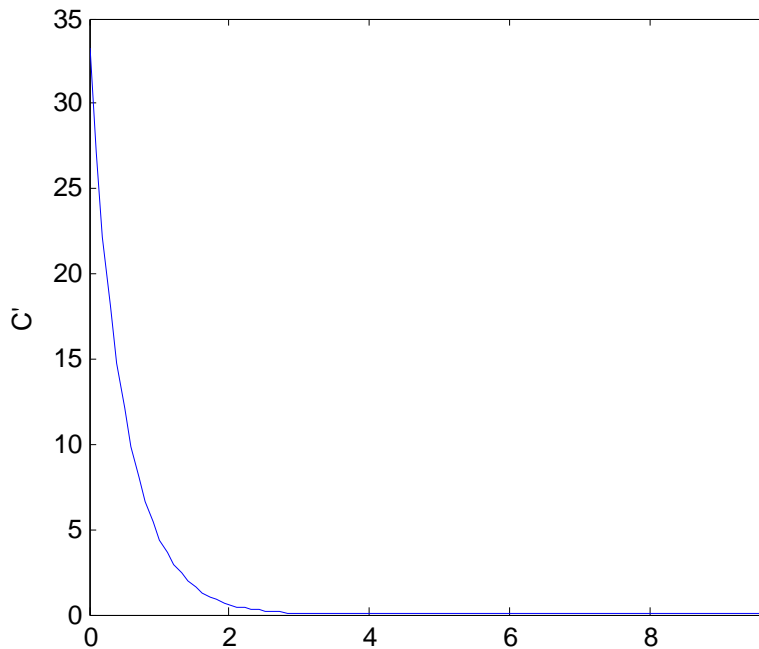


Figure 5.5 The C' field for the first sample point for the example function 1

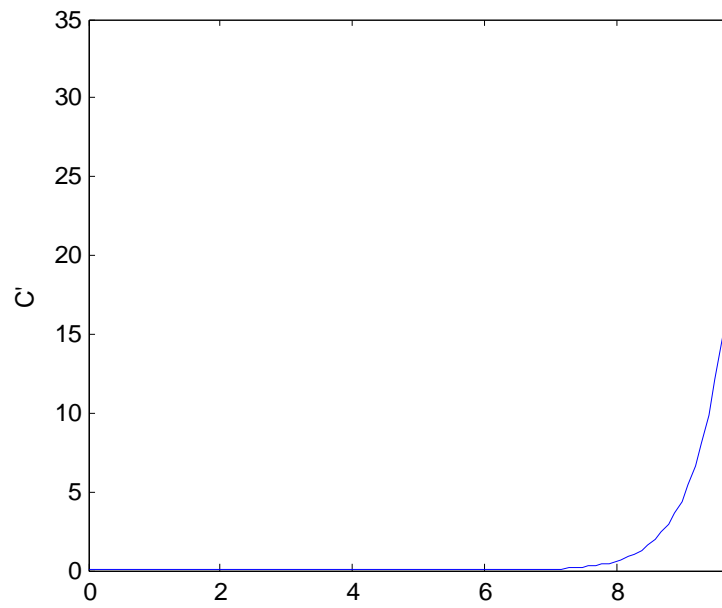


Figure 5.6 The C' field for the second sample point for the example function 1

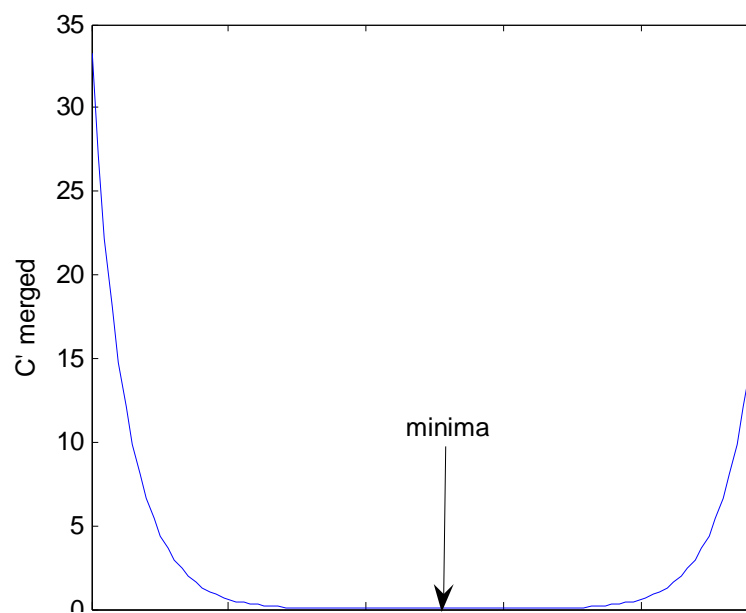


Figure 5.7 The C' field after merging and mirroring information from two sample point for the example function 1 and location of minima for the plot

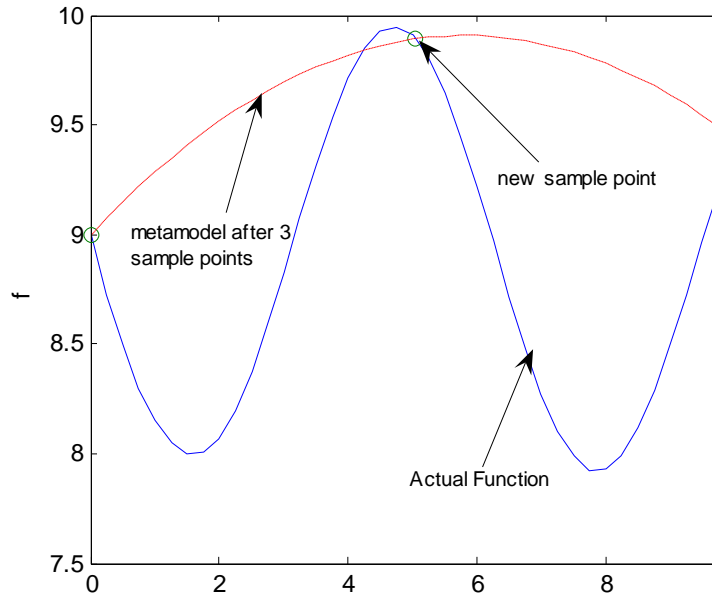
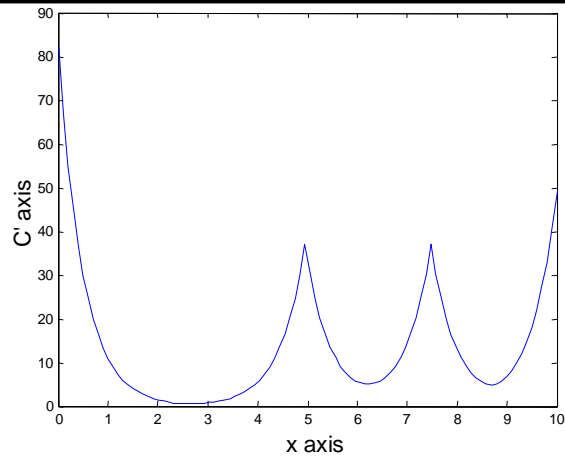
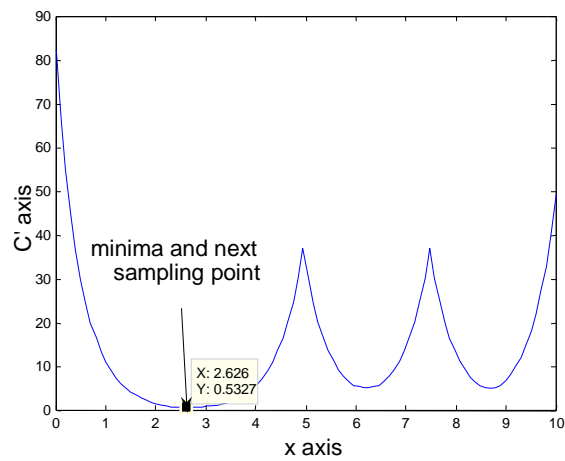


Figure 5.8 The metamodel after 3 sample points

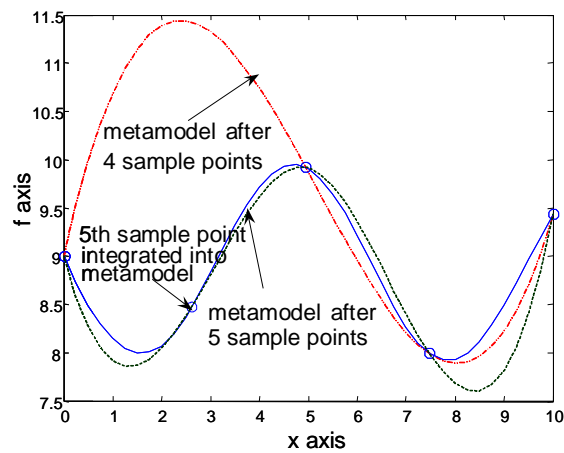
The output of Q2S2 methodology for the fifth iteration (adding the fifth sample point) is presented in Figure 5.9 for step 5 through step 7 of the Q2S2 flowchart (Figure 4.3). With 6 sample data points (Figure 5.10a), optimized as per the proposed methodology, the spline metamodel achieves more than a 95% correlation (same as Turner et al.) to the actual function while the Kriging metamodel achieves a correlation of 88%. With 10 optimized points (Figure 5.10b), all models achieve correlations greater than 99%. All models also predict optima within 2% of the actual optimal value of $x^* = 7.85$. It is important to note that in absence of any qualitative information the Q2S2 implementation samples at nearly uniform spaced intervals through out the space (just as sampling strategy with a Mean Squared Error or Maximum Likelihood Estimate does).



Step 5: C is created by summing or integrating along the f axis

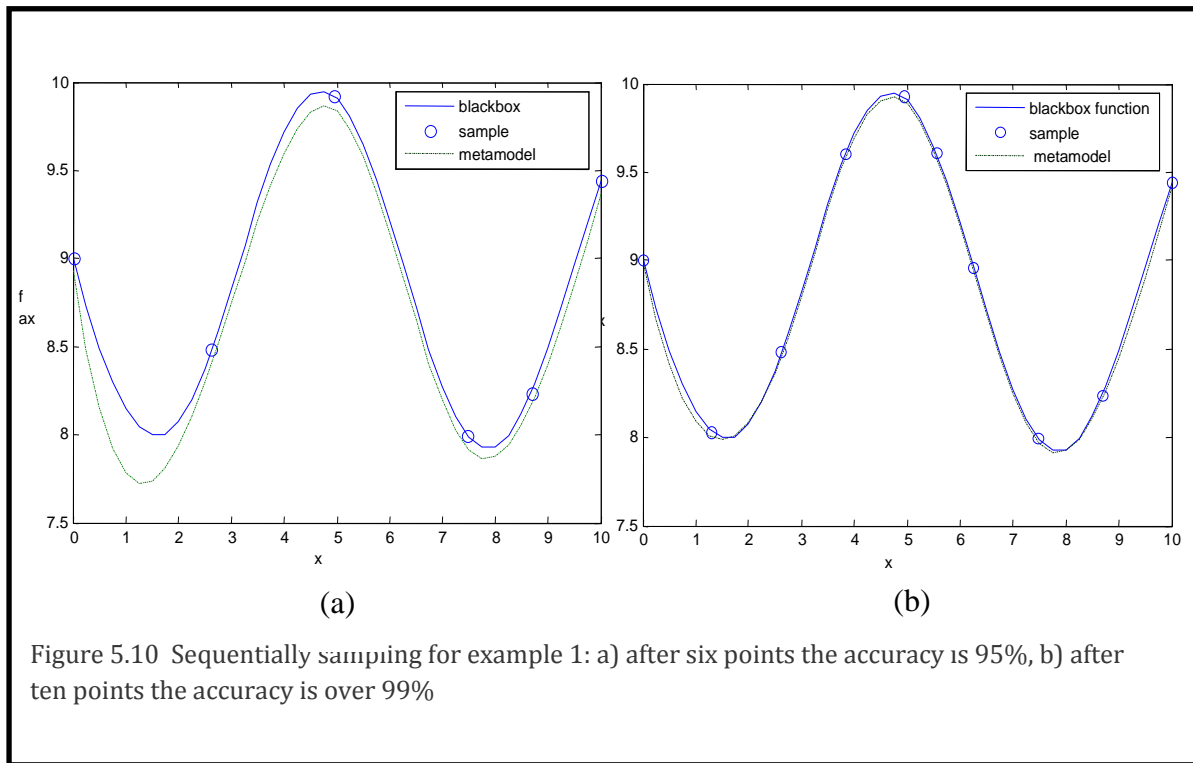


Step 6: optimization is used to find minimum of C



Step 7: metamodel is updated by sampling at this new data point

Figure 5.9 The step 5 through 7 of the Q2S2 process for adding 5th sample point



5.1.2. Example Function 2

Example function 2 was obtained from Martin and Simpson [2002]. This 1-D test problem models the output temperature of a chemical reaction. The input variable, ratio of oxidant to the fuel being burned, is increased from zero oxidant to an excess of oxidant. In this process, the reaction increases temperature (performance parameter) to a maximum and then decreases as excess oxidant is added as shown in Figure 5.11. The black-box model for this example is created by fitting temperature values, which are evenly spaced at 0.1 increments from 0 to 1 for a total of 11 points. The model used for fitting is a B-spline function available in Matlab.

For this example problem it is assumed that there is qualitative information available and hence the problem is more of a gray box as opposed to a black box where

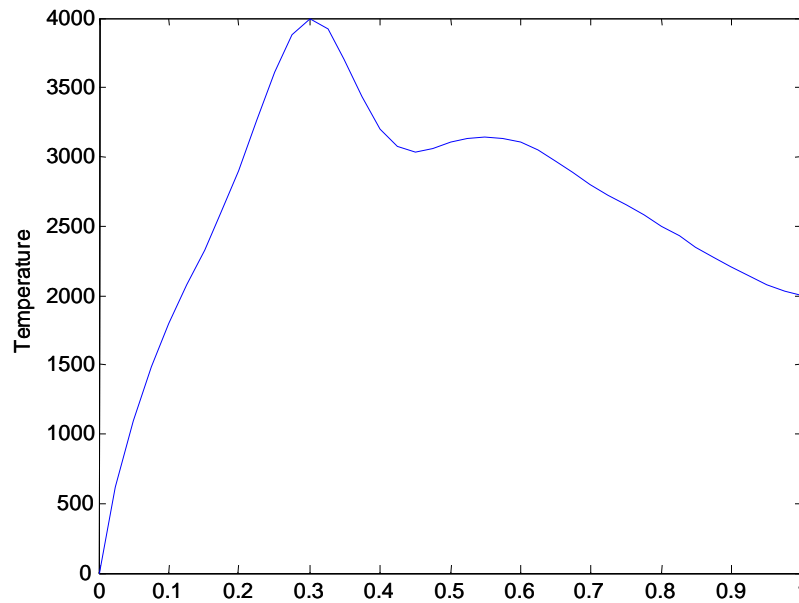


Figure 5.11 One Variable example function 2 taken from Martin and Simpson

```

version> MATLAB
Desktop Window Help
C:\MATLAB_SV701\work
What's New
<Student Version> Command Window
input
1. for new experimental data
2. for monotonicity information
3. when no more information available
1
input: x0 f0 ftol xtol in bracket i.e., [40 60 5 7] :[0.5 3000 300 0.1]
input
1. for new experimental data
2. for monotonicity information
3. when no more information available
EDU>> 2
input: sign x_min x_max ftol xtol in brackets i.e., [-1 40 60 5 7] :[1 .1 .3 200 .1]
input
1. for new experimental data
2. for monotonicity information
3. when no more information available
2
input: sign x_min x_max ftol xtol in brackets i.e., [-1 40 60 5 7] :[-1 .7 1 200 .1]
input
1. for new experimental data
2. for monotonicity information
3. when no more information available
3

```

Figure 5.12 The dialog between the Q2S2 process and the user currently takes place through a Matlab command window

no knowledge is known for the problem. Specifically, we have assumed that a user can provide two piecewise monotonicities. Clearly, the combustion temperature must rise initially as oxygen is introduced. Conservatively, we assume that between 0 and 0.3 the temperature is monotonically increasing. Furthermore, if the oxidant overwhelms the mixture it is likely that the combustion will in turn be reduced and so will the combustion temperature. Therefore, we believe it is safe to state that the function is monotonically decreasing in the range of 0.7 to 1.

As the Q2S2 process proceeds to Step 2, it asks the user to input the information that he/she may know the nature of the problem. This interaction involved in this step is illustrated in Figure 5.12 where the user's knowledge about the monotonicity of the problem is returned to the process in the form of a vector of values. Once the user has inputted the information, the program stores the information and if it needs further information it prompts the user. Once the user has no more information available the program continues with the next steps of Q2S2.

With 10 sample data points, optimized as per the proposed methodology the metamodel is shown in Figure 5.13. The metamodel has not quite captured the true nature of the problem in lower and higher values of x , as the number of sample points used in these regions is low. Conversely and interestingly the sampling strategy has been more effective in focusing on the region of the problem where the least information was known at the start and also where the optimum lies. This example illustrates that in presence of user qualitative knowledge proposed Q2S2 methodology has the ability to focus in regions different than usually predicted by totally quantitative sampling strategies. It also illustrates that with qualitative information the sampling strategies could be focused towards interesting regions (regions where the optimum of blackbox function lie) of the problem without wasting a lots of sampling points.

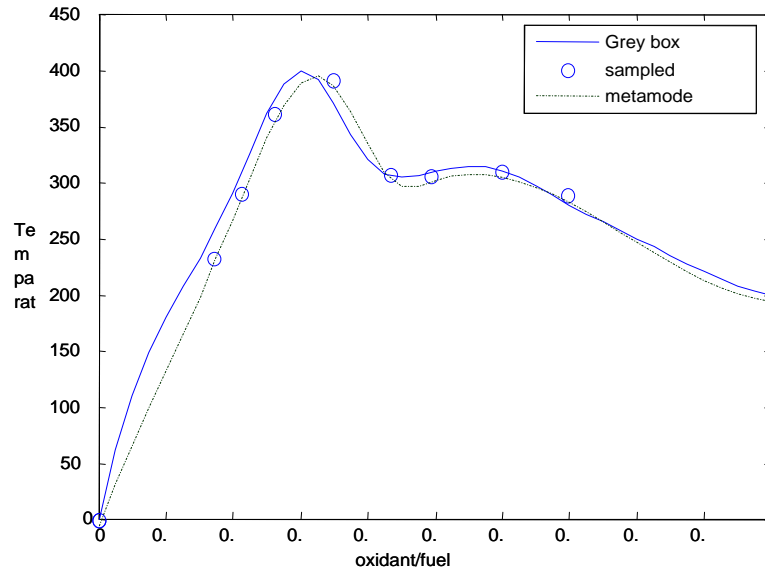


Figure 5.13 Sample points, actual grey box model and metamodel predicted by Q2S2 for example function 2

5.2. Multiple-Variable Functions and Comparison with Other Sampling Techniques

After successful implementation of the single variable function, a few multivariable functions were tested to ensure that the applicability is not reliant on a single variable. In this section, the performance of the Q2S2 technique by comparing its performance to that of several other classes of sampling plans found in the literature is also investigated. The sampling techniques used for comparison purposes are Full Factorial (FF) sampling, Latin Hypercube Design (LHD), and Hammersley Sequence Sampling (HSS) (Chapter 2). The performance measures used to compare the performance of the newly developed Q2S2 approach with that of the traditional sampling approaches is also presented in this section.

5.2.1. Metamodel Accuracy Measures

The overall performance of the sampling techniques is evaluated using six standard accuracy measures from literature [Simpson, 2001; Muller, 2005]. In order to measure the accuracy of the metamodels resulting from different sampling techniques, the standard error measure Root Mean Squared Error (RMSE) is used. *Small* values of the RMSE are desired. An RMSE value of 0 indicates a perfect fit. The error measure is defined as:

$$RSME = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - \tilde{f}_i)^2} \quad (5.1)$$

where, \tilde{f}_i is the corresponding predicted value for the observed value f_i . The n additional data points are evenly sampled in the design space. We use 1600 (40x40 grid for 2 variables) points for the two-variable problems (functions 1 to 3), 2187 (3x3x3x3x3x3x3 grid for 7 variables) points for the seven-variable problem (function 4 and 5). In addition to the above RMSE, we also calculate the Normalized Root Mean Squared Error (NRMSE) as follows:

$$NRSME = \sqrt{\frac{\sum_{i=1}^n (f_i - \tilde{f}_i)^2}{\sum_{i=1}^n f_i^2}} \times 100 \quad (5.2)$$

An NRMSE value of 0 indicates a perfect fit. Third measure is R square, we calculate the R square as follows:

$$RSQUARE = 1 - \frac{\sum_{i=1}^n (f_i - \tilde{f}_i)^2}{\sum_{i=1}^n (f_i - \bar{f})^2} = 1 - \frac{MSE}{VARIANCE} \quad (5.3)$$

where, \bar{f} is the mean of the observed values. While MSE (Mean Square Error) represents the departure of the metamodel from the real simulation model, the variance captures how irregular the problem is. The *larger* the value of R Square, the more accurate the metamodel. The Fourth measure is Relative Average Absolute Error (RAAE):

$$RAAE = \frac{\sum_{i=1}^n |f_i - \tilde{f}_i|}{n \cdot STD} \quad (5.4)$$

where, STD stands for standard deviation. The *smaller* the value of RAAE, the more accurate the metamodel. Fifth measure is Relative Maximum Absolute Error (RMAE):

$$RMAE = \frac{\max(|f_1 - \tilde{f}_1|, |f_2 - \tilde{f}_2|, \dots, |f_n - \tilde{f}_n|)}{STD} \quad (5.5)$$

A *small* RMAE is preferred. Sixth measure is Maximum Absolute Bias (MAXAB):

$$MAXAB = \max(|f_1 - \tilde{f}_1|, |f_2 - \tilde{f}_2|, \dots, |f_n - \tilde{f}_n|) \quad (5.6)$$

A *small* MAXAB is preferred. Large RMAE and MAXAB values indicates large error in one region of the design space even though the overall accuracy indicated by RSME, NRSME, R Square and RAAE can be very good. However, since RMAE and MAXAB metric cannot show the overall performance in the design space, they are not as important as RSME, NRSME, R Square and RAAE. Only higher value of R Square is desirable, while smaller values of RSME, NRSME, RMAE, RAAE and MAXAB are sought.

5.2.2. Sample Size and Metamodel

The effect of sample size on metamodel accuracy is an important criterion for selecting an appropriate sampling approach. Typically, for interpolative metamodels, the higher the number of samples, the higher is the metamodel accuracy. For the test function in this paper, we select sample size based on two criteria. First, the sample size should correspond in number to a full factorial design (n^k). Secondly, if the first criterion is satisfied the sample size should be less than ten times the number of variables in the problem. This results in 16 sample points for 2 variable problems and 128 sample points for 7 variable problems. After the sample point generation a blackbox model for each of

Name	Function	No. of variables	Design Domain
Test function 1 (Projectile)	$2v_0^2 \sin \varphi \cos \varphi / g$	2	$0 \leq v_0 \leq 18$ $0 \leq \varphi \leq 90$
Test function 2	$x_1 \sin(x_1) + x_2 \sin(x_2)$	2	$0 \leq x_i \leq 2$
Test function 3	$[30 + x_1 \sin(x_1)] * [4 + \exp(-x_2^2)]$	2	$0 \leq x_1 \leq 10$ $0 \leq x_2 \leq 6$
Test function 4	$(x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$	7	$0 \leq x_i \leq 10$
Test function 5	$\{2 T_u (H_u - H_l)\} / \{\ln(r/r_w) [1 + (2LT_u / \ln(r/r_w) r_w^2 K_w) + T_u / T_l]\}$	7	$0 \leq r_w \leq 10$ $100 \leq r \leq 50,000$ $0 \leq T_u \leq 115,600$ $900 \leq H_u \leq 1110$ $63.1 \leq T_l \leq 116$ $700 \leq H_l \leq 820$ $1120 \leq L \leq 1680$ $K_w = 9855$

Table 5.1 Multivariable Test Problems

Name	Qualitative information assumed
Test function 1 (Projectile)	$\varphi [0 \ 30]^+ \ \varphi [60 \ 90]^-$
Test function 2	$x_1 [0 \ 4]^+, x_1 [5 \ 4 \ 3 \ 2]^- , x_2 [0 \ 4]^+, x_2 [5 \ 4 \ 3 \ 2]^-$
Test function 3	$x_1 [0 \ 3]^+, x_1 [9 \ 10]^+, x_2 [5 \ 6]^-$
Test function 4	$x_3 [0 \ 3]^+, x_3 [9 \ 10]^+, x_5 [0 \ 3]^+, x_5 [9 \ 10]^+$
Test function 5	$H_u [1050 \ 1110]^+, H_l [700 \ 730]^- , L [1120 \ 1220]^-$

Table 5.2 Assumed Monotonicity Information for the Test Problems

these sampling strategies was formulated through the Kriging metamodel within the DACE toolbox of Matlab (Appendix A).

5.2.3. Multiple-Variable Functions

Based on the proposed scheme for comparative study, kriging metamodels are created for the 5 test problems [Palmer 2001, Simpson, 2001; Muller, 2005] (4 from literature), using different sampling techniques (see Table 5.1). Table 5.2 presents

assumed qualitative information about the problems that has been used for Q2S2 method. For comparison purposes we also present the performance of Q2S2 methodology in absence of any qualitative information (represented as QS in plots). Different techniques are compared based on the results from different metamodel accuracy measures. Boxplot graphs are used for comparisons. The boxplots depict the high, low, and median values, as well as the quartiles, for sampling techniques having multiple plans (LHD). A '+' denotes an outlier. For sampling techniques having a deterministic plan, a horizontal line shows the value of the measure for that plan.

Figure 5.14 to 5.18 shows the comparison for different sampling techniques. Each figure displays the six metamodel accuracy measures for a given test function. In absence of any qualitative information Q2S2 methodology (represented as QS in box plots) performs very close to HSS plans. This may be due to the reason that in absence of any qualitative information the Q2S2 tries to sample the design space in a uniform manner.

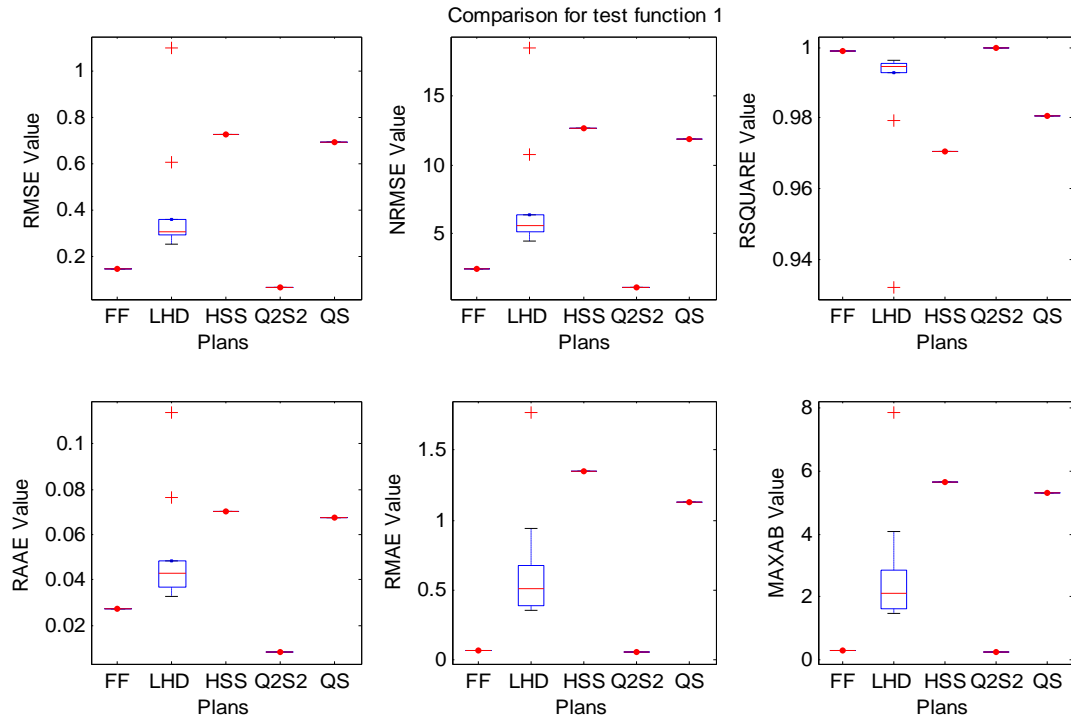


Figure 5.14 Box plots for multivariable test function 1

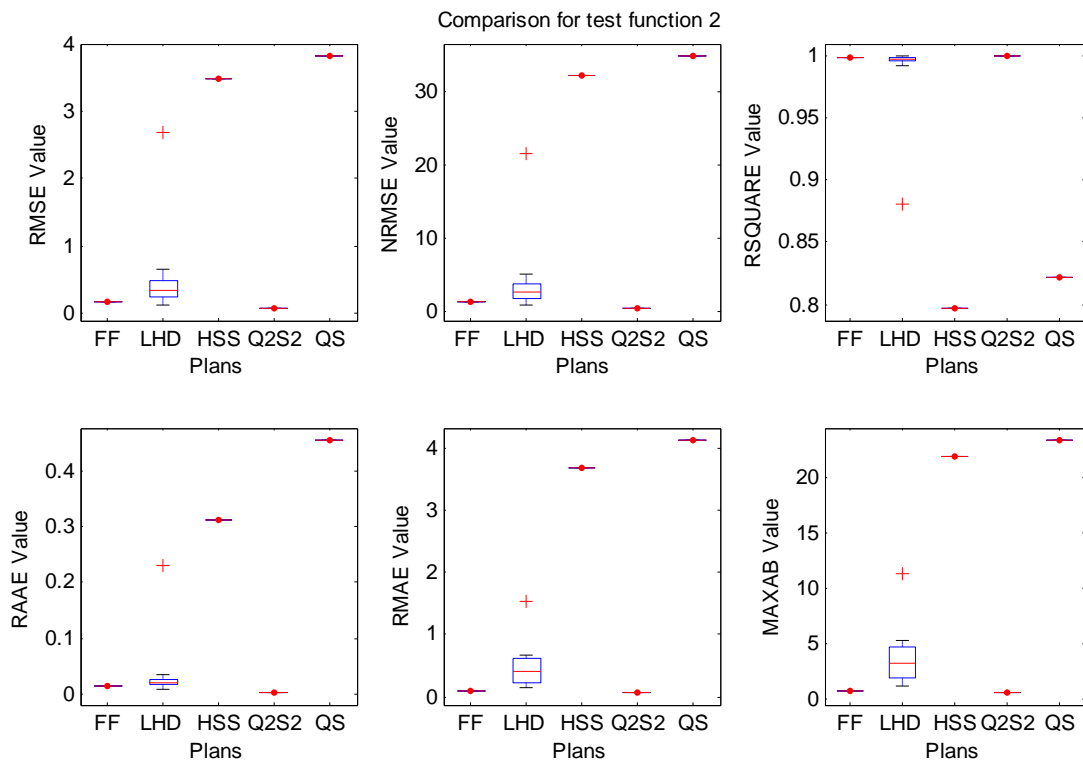


Figure 5.15 Box plots for multivariable test function 2

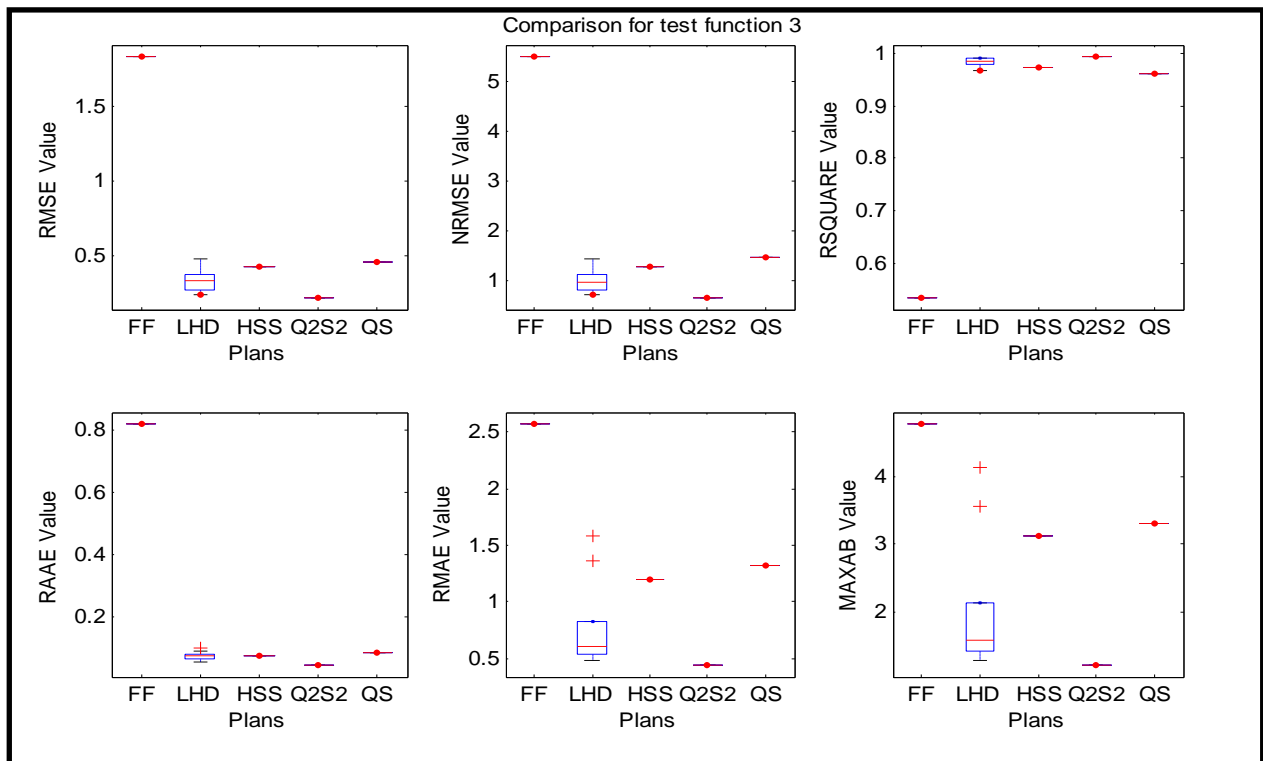


Figure 5.16 Box plots for multivariable test function 3

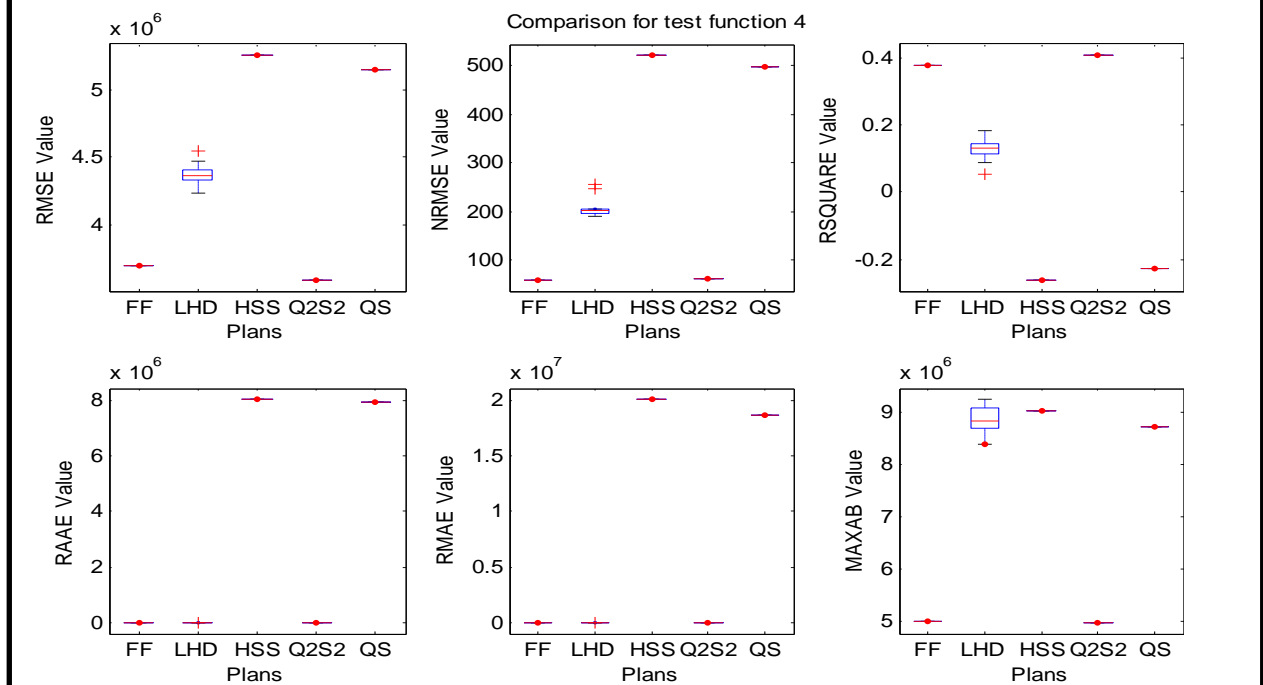
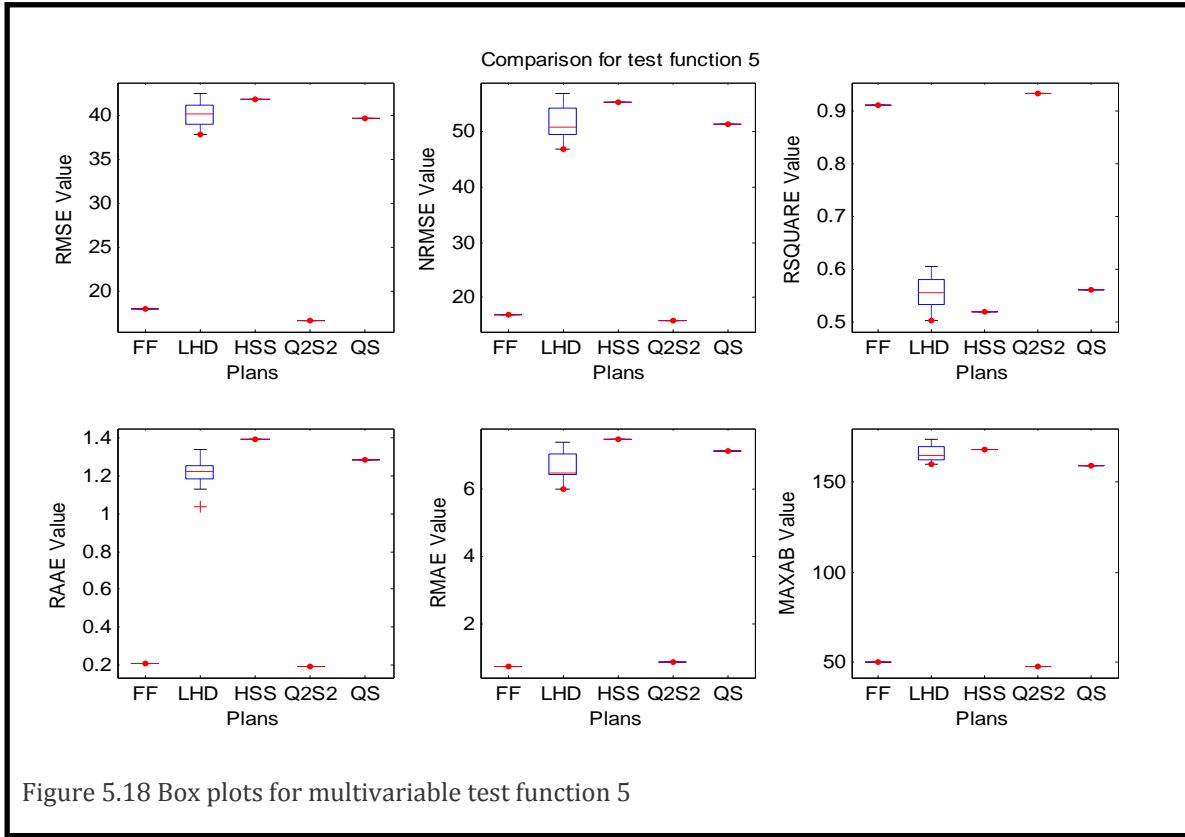
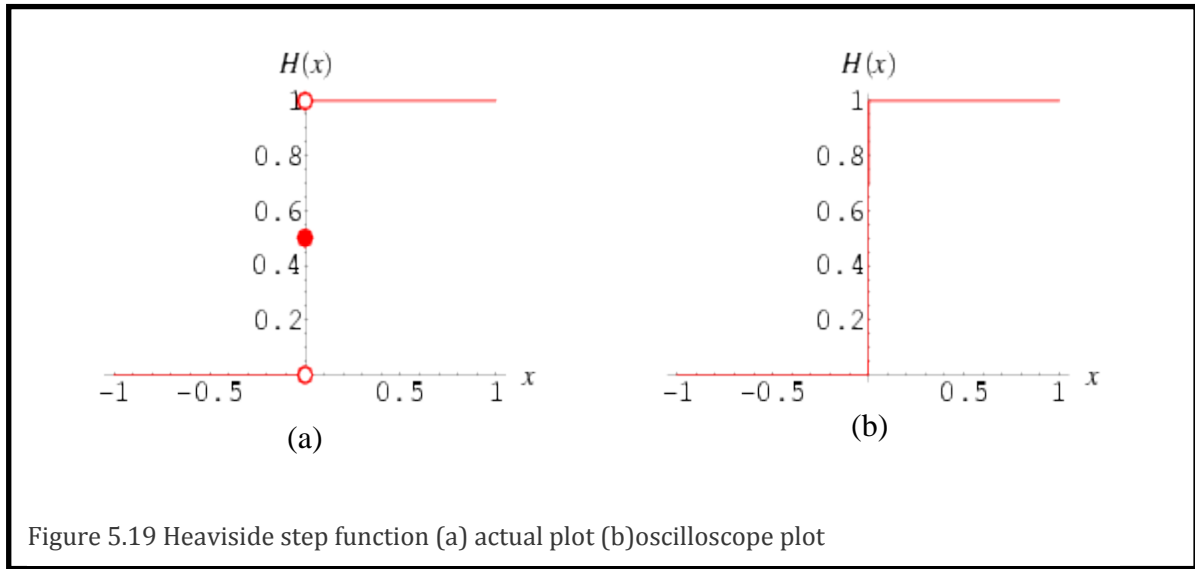


Figure 5.17 Box plots for multivariable test function 4



It is noteworthy that even in absence of qualitative information the QS performs significantly well, in some cases (test function 1 and 3) it performs better than some of the LHD plans. However, the median of comparison measures for the LHD plans are better than QS plans.

However once the qualitative information about the problems are supplied to Q2S2, the performance of the Q2S2 increases significantly. For each of the test function s shown the Q2S2 produces the best sample points in light of qualitative information! As the number of variables increases Q2S2 is clearly better than LHD and HSS sampling techniques in every measure. For seven variable problems FF sampling is closest to Q2S2 in RMAE and MAXAB measures but as discussed earlier these measures are local measure, and Q2S2 performs better in other measures such as RSME, NRSME, and R square which are global measures.



5.2.4. Discontinuous Function

These previous example problems are continuous functions. In order to test the effectiveness of the Q2S2 methodology with respect to discontinuous function, a Heaviside step function was used as a test bed. The Heaviside step function $H(x)$ is a discontinuous function also known as the "unit step function" and defined by

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 0.5 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \quad (5.7)$$

The plots above show the function as defined (Figure 5.19 a), and how it would look if displayed on an oscilloscope (Figure 5.19 b). The output of Q2S2 methodology for 11 sample points in absence of any qualitative information (QS) is presented in Figure 5.20. For Q2S2 purpose, it is assumed that a user can provide two piecewise monotonicities. It is assumed that between the range -1 and -0.4 and the range 0.4 to 1.0 the function is constant. The output of Q2S2 methodology for 11 sample points in presence of this set of qualitative information is presented in Figure 5.21.

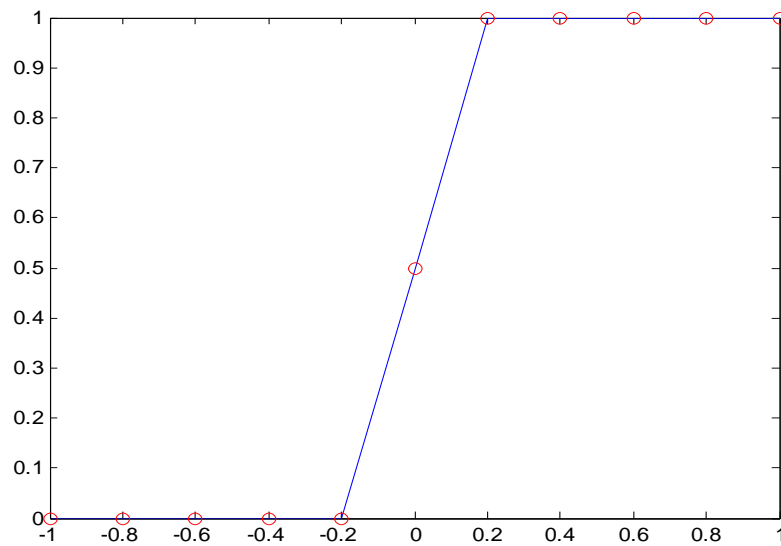


Figure 5.20 The metamodel and sample points for Heaviside step function after 11 sample points in absence of any qualitative information

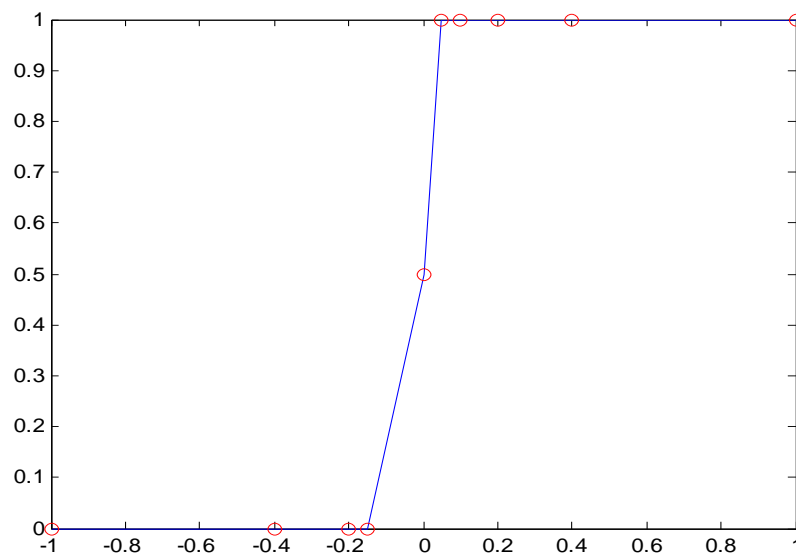


Figure 5.21 The metamodel and sample points for Heaviside step function after 11 sample points in presence of first set of qualitative information

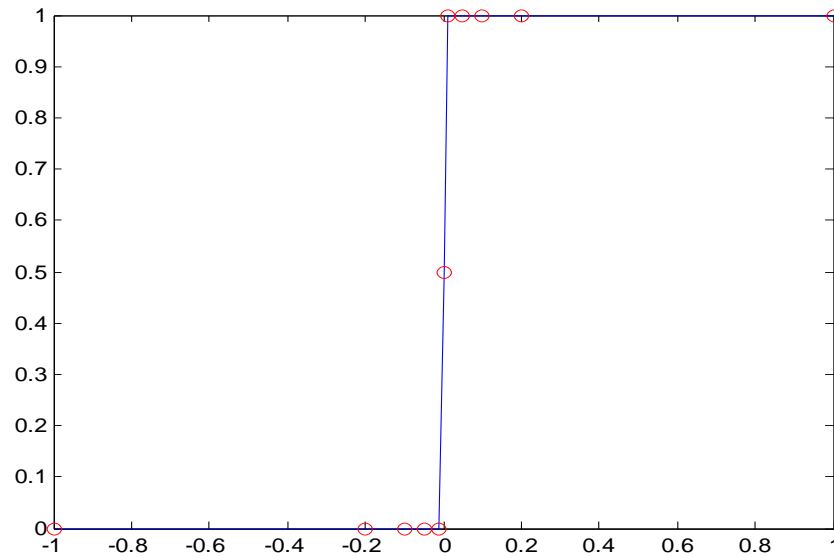


Figure 5.22 The metamodel and sample points for Heaviside step function after 11 sample in presence of second set of qualitative information

Furthermore, a second set of qualitative information was also considered. It is assumed that between the range -1 and -0.2 and the range 0.2 to 1.0 the function is constant. The output of Q2S2 methodology for 11 sample points in presence of this set of qualitative information is presented in Figure 5.22. The RSME values for the metamodels in Figure 5.20, 5.21, and 5.22 are 0.19, 0.08, and 0.03 respectively. The RSQUARE values for the metamodels in Figure 5.20, 5.21, and 5.22 are 0.87, 0.94, and 0.98 respectively. It can be concluded that the Q2S2 methodology can predict sampling points to accurately model the discontinuous Heaviside function in light of appropriate qualitative information. The reader is reminded that Q2S2 only determines the choice of sample points while a Kriging model is used to arrive at the metamodel.

5.3. Tuning the “knobs”

In this section a guidance strategy to help the user of Q2S2 is provided. The intent of this section is twofold. First, it is compared to show how different information storing

			f_{tol}, x_{inf}								
			0.0	0.05	0.1	.15	.20	.25	.3	.35	.4
Exponential	Test	RSME	0.23	0.17	0.09	0.17	0.27	0.35	0.42	0.54	0.57
		Rsquare	0.81	0.85	0.92	0.85	0.81	0.69	0.63	0.61	0.58
	Func1	RSME	0.31	0.15	0.08	0.13	0.34	0.47	0.42	0.44	0.48
		Rsquare	0.74	0.94	0.93	0.91	0.77	0.62	0.59	0.53	0.47
	Test	RSME	0.29	0.18	0.17	0.13	0.31	0.38	0.48	0.49	0.51
		Rsquare	0.76	0.89	0.92	0.87	0.78	0.74	0.68	0.61	0.58
General Exponential	Test	RSME	0.24	0.16	0.07	0.16	0.24	0.34	0.41	0.55	0.59
		Rsquare	0.83	0.87	0.91	0.95	0.83	0.68	0.67	0.66	0.57
	Func1	RSME	0.36	0.16	0.09	0.11	0.36	0.51	0.41	0.47	0.47
		Rsquare	0.71	0.91	0.91	0.94	0.75	0.64	0.62	0.51	0.48
	Test	RSME	0.27	0.16	0.15	0.12	0.33	0.33	0.52	0.57	0.61
		Rsquare	0.78	0.87	0.94	0.89	0.80	0.79	0.71	0.64	0.58
Gauss	Test	RSME	0.26	0.18	0.08	0.16	0.28	0.37	0.45	0.53	0.57
		Rsquare	0.78	0.87	0.94	0.89	0.79	0.71	0.68	0.61	0.59
	Func1	RSME	0.28	0.13	0.06	0.11	0.31	0.43	0.47	0.49	0.52
		Rsquare	0.77	0.92	0.98	0.94	0.74	0.67	0.61	0.58	0.54
	Test	RSME	0.31	0.17	0.11	0.10	0.29	0.38	0.46	0.51	0.53
		Rsquare	0.79	0.85	0.97	0.89	0.81	0.72	0.69	0.63	0.57
Linear	Test	RSME	0.27	0.18	0.10	0.19	0.27	0.37	0.46	0.58	0.61
		Rsquare	0.81	0.83	0.87	0.89	0.81	0.66	0.64	0.61	0.63
	Func1	RSME	0.39	0.19	0.13	0.17	0.38	0.40	0.43	0.51	0.47
		Rsquare	0.76	0.87	0.86	0.86	0.71	0.62	0.59	0.57	0.52

	Test	RSME	0.29	0.19	0.16	0.18	0.33	0.38	0.48	0.54	0.64
	Func3	Rsquare	0.76	0.82	0.90	0.83	0.87	0.81	0.76	0.69	0.54
Spherical	Test	RSME	0.21	0.17	0.11	0.17	0.27	0.36	0.46	0.52	0.58
	Func1	Rsquare	0.81	0.89	0.91	0.88	0.81	0.73	0.66	0.63	0.58
	Test	RSME	0.25	0.12	0.08	0.13	0.32	0.41	0.49	0.47	0.51
	Func2	Rsquare	0.83	0.94	0.96	0.91	0.72	0.70	0.59	0.56	0.51
	Test	RSME	0.27	0.18	0.14	0.12	0.31	0.37	0.45	0.53	0.48
	Func3	Rsquare	0.82	0.84	0.92	0.87	0.85	0.74	0.67	0.62	0.53
Cubic	Test	RSME	0.22	0.16	0.09	0.17	0.26	0.35	0.44	0.52	0.63
	Func1	Rsquare	0.86	0.85	0.91	0.90	0.83	0.69	0.66	0.66	0.61
	Test	RSME	0.34	0.18	0.11	0.16	0.36	0.37	0.41	0.55	0.49
	Func2	Rsquare	0.81	0.89	0.89	0.88	0.77	0.67	0.63	0.60	0.49
	Test	RSME	0.24	0.15	0.13	0.16	0.24	0.33	0.38	0.51	0.61
	Func3	Rsquare	0.81	0.92	0.94	0.87	0.85	0.79	0.72	0.68	0.52
Spline	Test	RSME	0.24	0.16	0.09	0.18	0.25	0.34	0.47	0.56	0.52
	Func1	Rsquare	0.76	0.88	0.93	0.89	0.80	0.77	0.64	0.59	0.53
	Test	RSME	0.29	0.11	0.08	0.12	0.31	0.39	0.48	0.43	0.49
	Func2	Rsquare	0.78	0.93	0.96	0.94	0.74	0.74	0.62	0.59	0.53
	Test	RSME	0.31	0.16	0.13	0.17	0.33	0.35	0.49	0.51	0.46
	Func3	Rsquare	0.76	0.87	0.94	0.87	0.79	0.77	0.72	0.67	0.51

Table 5.3 Bi-stable structure design variables and associated monotonicity information

models (Table 4.1) behave. Secondly, appropriate values for the knobs of the model i.e., f_{tol} and x_{inf} is found out. Test function 1, 2 and 3 from Table 5.1 were used as test bed function for these purposes.

Due to unavailability of suitable optimization methods and computationally prohibitive evaluations involved in step 6 of Q2S2 methodology (Figure 4.4) for information storing models other than the gaussian model, only two dimensional problems were considered. Among the six different metrics of comparison only RSME and RSQUARE metrics were selected to proceed further. The main reason for this choice is similarity in the behavior of RSME and other metrics like NRSME, RAAE, RMAE, and MAXAB. The ranges of the variables were normalized to get a range of 0 to 1 for each variable. The knobs of the model i.e., f_{tol} and x_{inf} were incrementally varied 0.05 to 0.4 at step of 0.05 for each different information storing models. Values greater than 0.4 results in higher error and hence omitted from this study. The results of this exercise is presented in Table 5.3. It can be concluded that there is less than 10% variation in both metric values for same test problem and different information storing models. Hence, it can be concluded that every information storing models performs nearly the same. However, due to the ease in optimization afforded by gaussian model, the gaussian model is recommended to the user of Q2S2. Even though every model performs nearly same for 2 variable problem, the Q2S2 implementation for higher dimension problems will be difficult with models other than gaussian model. Secondly, as can be seen from Table 5.3 the values for knobs of the model i.e., f_{tol} and x_{inf} that give the best results are in the range of 0.05 to 0.15 of the normalized value of the variables (as highlighted in green in Table 5.3).

5.4. Recapitulation

The examples presented in this chapter are valuable from the point of view of having a means to better understand of the Q2S2 process. It also provides insights into the behavior of Q2S2 methodology and compares its performance against currently available sampling techniques. A application of Q2S2 methodology on a single variable function is indeed the most useful in such a case, because all relevant behavior can be seen in one dimension, and this aids the visualization tremendously. Once the salient aspects of the Q2S2 methodology were illustrated through one variable example problems, the superior performance of Q2S2 methodology was also demonstrated through 5 different multivariable problems. Now having a feel for how to use or apply such a technique in general, the next step is to implement this method in an engineering setting. The next chapter details how Q2S2 can be used in design of Bi-Stable MEMS devices.

Chapter 6

Q2S2 Implementation for Design of Bi-Stable MEMS Relay

This chapter demonstrates a step-by-step implementation of the Q2S2 methodology to a mechanical engineering design problem. This is to ensure that the applicability of the proposed process is not purely theoretical, but will also stand up to engineering scrutiny in a complex real-world applications.

6.1. Introductory Comments and Motivation Behind Bi-Stable MEMS Relay

6.1.1. Multistable Equilibrium Systems

Multistable equilibrium (MSE) systems are those which have more than one stable equilibrium position or configuration [King, 2004]. Each stable equilibrium is defined as the state in which the system has a minimum in its potential energy. An unstable equilibrium is defined as a state in which the system has a maximum in its potential energy.

Multistable equilibrium systems bridge the gaps between adaptive structures and their limited range and energy inefficiency. MSE structures will have a wide range of operating regimes. Each range will lie between the stable equilibrium states. As the number of stable equilibrium positions increase, the operating range of the structure increases. The key aspect of the MSE systems is that they can act as passive structures for the vast majority of time, only requiring actuation to move among stable positions or about the current equilibrium. This will reduce the power consumption of the system and improve the reliability since the system will have low potential energy before and after actuation.

MSE systems are highly non-linear. The mathematical understanding behind these systems is based only on the potential energy curve. There is no established methodology for designing such systems. At the same time, the properties of MSE systems are desirable in applications such as optical switches, relays and reconfigurable structures where low power consumption and reliability are key concerns. Thus, MSE systems introduce a new design philosophy in adaptable, or reconfigurable, structures.

6.1.2. Micro-Electro Mechanical (MEMS) Relays

Relays play an important role in a wide range of applications in automobiles, telecommunications, measurement equipment, automatic test equipment, robotics and cell phones. Electromechanical relays are still widely used despite the advantages of solid-state relays which are smaller in size and can be batch produced [Taylor, 1998]. This is because the solid-state relays suffer from poor performance characteristics like maximum off-state resistance, and higher contact power dissipation as compared to their electromagnetic counterparts. Micromachined relays are produced through the chemical vapor deposition process used to make integrated circuits. This approach attempts to combine the best attributes of both electromechanical relays and solid-state relays, namely, the smaller size of solid-state devices and the increased off resistance and lower on resistance that is more typical of macroscopic electromechanical relays. Micro-machined relays may also allow for the interconnection of large relay arrays during fabrication, thus reducing other fabrication steps currently needed for conventional relays.

A bi-stable MEMS relay will have a large number of applications and advantages over conventional electromechanical relays. This chapter presents design methodology to synthesize MEMS relays based on Q2S2 methodology that searches for the best topologies for the desired bi-stable equilibrium characteristics.

6.1.3. Understanding Characteristics of MSE Systems

Equilibrium in a mechanical system is a configuration or state where all the forces and torques acting on it are balanced. The resultant force on the system is zero in such equilibrium positions. The system can remain in the state of equilibrium until acted upon by external forces. Equilibria (states of equilibrium) may be unstable, neutral or stable. An unstable equilibrium requires hardly any disturbance to be lost. A neutral equilibrium is arbitrary or non-preferential, whereas a stable equilibrium is such that a system returns to it after the displacing disturbance is removed.

Each stable position requires no power input to maintain the position, and the minimization of dissipative forces can make the stable positions very accurate and repeatable. A bi-stable system is a system having two stable equilibrium positions.

The equilibrium characteristics of a multistable compliant structure can be described by its potential energy curve. The basic design methodology of the MSEs is based on the fact that the potential energy of the structure is at a minimum and at a maximum at stable and unstable positions, respectively (Figure 6.1).

Although the potential energy-displacement curve provides an intuitive understanding of MSE systems, it is difficult to calculate the potential energy of a system during analysis. On differentiating the potential energy we get the force on the system. As the force (or reaction force) is easier to calculate for a particular device, the force-displacement curve is generally preferred to identify multi-stability of a system.

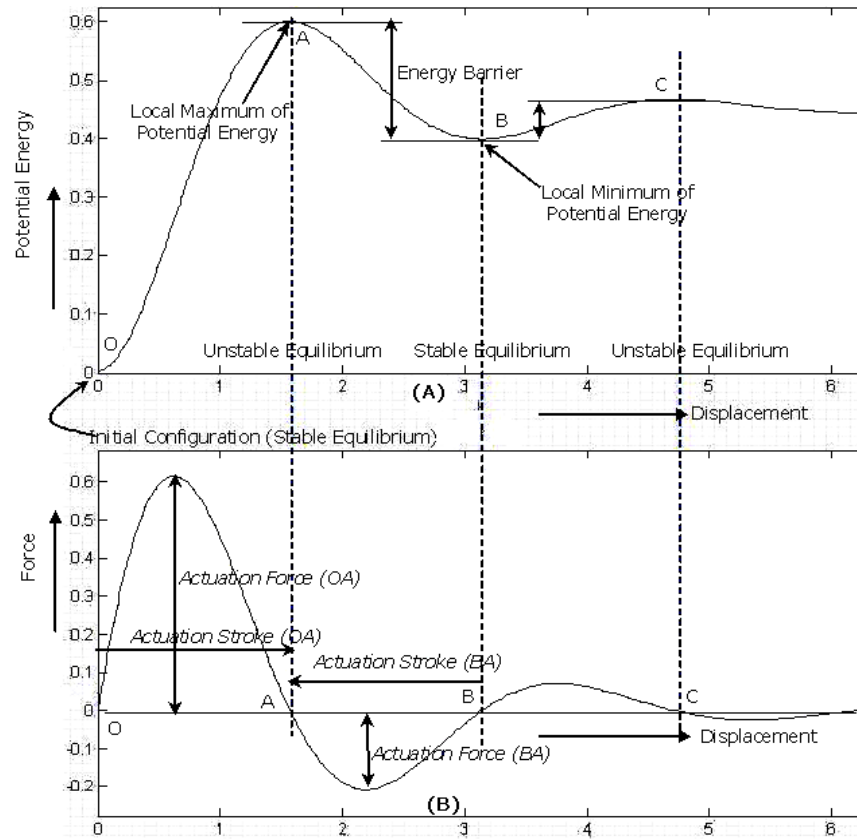


Figure 6.1 A) Potential Energy as a function of changing configuration of a MSE system B) Force vs Displacement curve of a MSE system [King, 2004]

Multi-stable systems can be understood from the force displacement curve. Further, the force-displacement curve is also helpful in identifying other design parameters such as actuation force and actuation stroke length. Figure 6.1 relates the potential energy and force-displacement curve and shows some relevant terms.

The positions of the stable equilibrium, maximum actuation force and the actuation stroke length are the primary functional design parameters of the MSE system. The design should also include the material constraints so that the stresses generated by the operating loads stay below the yield stress. Apart from these constraints, geometric or

topology parameters such as size of the structure, number of beams and their dimensional constraints are defined separately.

6.1.4. Representation of Complaint Structures

The MSE compliant structures are made of beam and anchor elements. To represent such a system we first create a model of a beam, which can then be used to represent each beam in the structure. The beam model is defined as:

$$Beam \quad l \quad w \quad (6.1)$$

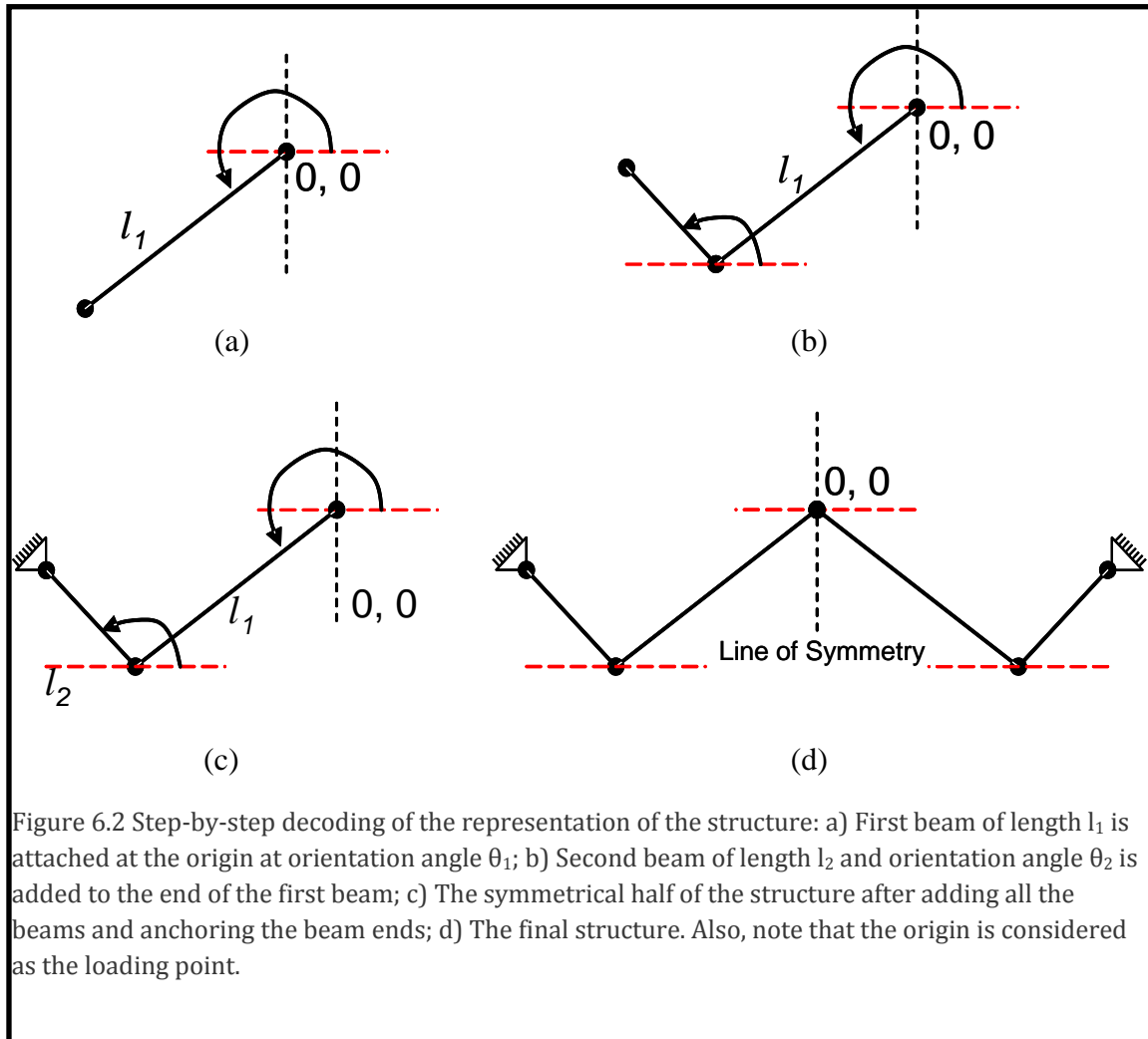
The design variables include the dimensions of the beams and their orientations in 2-D space. Each beam is assumed to be initially straight, have uniform cross-section and no initial stresses.

Once the beams and their connectivity are defined, the complete structure for two beams can have the following representation:

$$Structure \quad Beam_1 \quad Beam_2 \quad (6.2)$$

Each of the design variables in the beam has an upper and lower bound that aids in establishing the topology. The first beam of the structure is always positioned at the origin. The second beam is then connected to the end point of the first beam. The open ends of the 2nd beam which is not inter-connected to the structure is anchored to ground and assumed to have zero degrees of freedom. Figure 6.2 is a step by step example of building a structure of two beams.

Since the structure of Figure 6.2c is symmetric in nature, the representation only has one half of the complete structure. The other half can be obtained taking its mirror image along the line of symmetry. We also assume that the KP = 0 is the loading point of the structure. For simplicity, the loading is always along the line of symmetry in the positive y-axis. The symmetry of the structure further implies that the forces along the x-



axis at the origin are balanced. In fully-compliant systems, all the links do not provide a relative rotation like pin joints and hence the moments at all joints including the origin are also assumed to be internally balanced.

6.2. Q2S2 Based Solution Approach

The model of the structure generated through the representation method is tested through ANSYS program. The model of the structure is appropriately meshed, and the displacement load is applied on the loading point along the y-axis in small increments

from the initial position (i.e. the origin) to the desired second stable equilibrium position. This strategy of loading is specific for the design of bi-stable compliant systems but is extendable to higher order multistable systems. A nonlinear FEM analysis is performed on the model, and the output obtained is the reaction force on the loading point for each incrementally applied displacement.

The Delta-Slope method is defined to capture this bi-stable characteristic. An observation of the force-displacement plot for a bi-stable system is that the slope of the force reduces the more the structure deflects.

$$fDS = \text{sgn}(\text{force}) * \frac{d^2}{dx^2} \text{force} \quad \#(\text{sign changes}) \quad (6.3)$$

The delta-slope method is actually an approximation of the second-derivative of the force with respect to the displacement, x . When the force is positive, we want a large negative second derivative so that it will approach the x -axis. When the force is negative, we reward it when the second derivative is positive (Figure 6.1). Thus we have to design a device which has a spring softening quality instead of the typical spring hardening quality when the system is deflected.

For further details pertaining to the MSE compliant structure readers are referred to following papers [Eashwer, 2004]. This problem is highly nonlinear in nature has been solved previously by the application of genetic algorithms [Eashwer, 2004]. The objective here in this paper is to generate a feasible (not an optimal) solution with minimum number of FEA evaluations. The feasible solutions exhibit the property of bi-stability. The design variables and their range for the problem are presented in Table 6.1.

In order to generate qualitative information about the problem, two Macro level prototypes were manufactured using SLS (Figure 6.3). These prototypes were provided to a group of seven graduate students (ADLab UT) in brainstorming session for generating

Design Variables	Monotonic Information	Significance
$\theta_1, \theta_2, l_1, l_2, w_1, w_2$	$\theta_1[0, \pi/6]$ $\theta_1[3\pi/8, \pi/2]$ $\theta_1[7\pi/4, 2\pi]$ $\theta_1[5\pi/4, 3\pi/2]$	This orientation of beam 1 produces “M” shape which is desirable for spring softening
Variable Range and Problem Constants	$\theta_2[\pi/2, \pi]$, $\pi/2]$ $\theta_2[\pi/2, \pi/2]$	This orientation of beam 2 produces enough supporting force to push device past unstable equilibrium position
$100 \leq l_1 \leq 300$ $10 \leq l_2 \leq 50$ $0 \leq w_1 \leq 2$ $0 \leq w_2 \leq 2$ $5 \leq w_1 \leq 10$ $h_1 \leq h_2 \leq 3$ Yield Strength=7 GPa Poisson's Ratio=0.25 Young Modulus =169 GPa All the dimensions are in micrometers	$l_1[200,300]$ $l_2[10,30]$ $\theta_1[\sin^{-1}((A_l/2)/l_1)]$ $\theta_1[\sin^{-1}((A_l/2)/l_1)]$	The larger the length of beam 1 more compliant it is (deflection theory). The shorter the beam 2 the higher the force generated in the device to overcome unstable equilibrium position This orientation of beam 1 has desirable property of achieving the desirable Actuation length
	Where .3, + and – indicative type of monotonicity	

Table 6.1 Bi-stable structure design variables and associated monotonicity information

qualitative information. The qualitative information generated and their significance is presented in Table 2. The maximum number of sample point provided to Q2S2 was 50

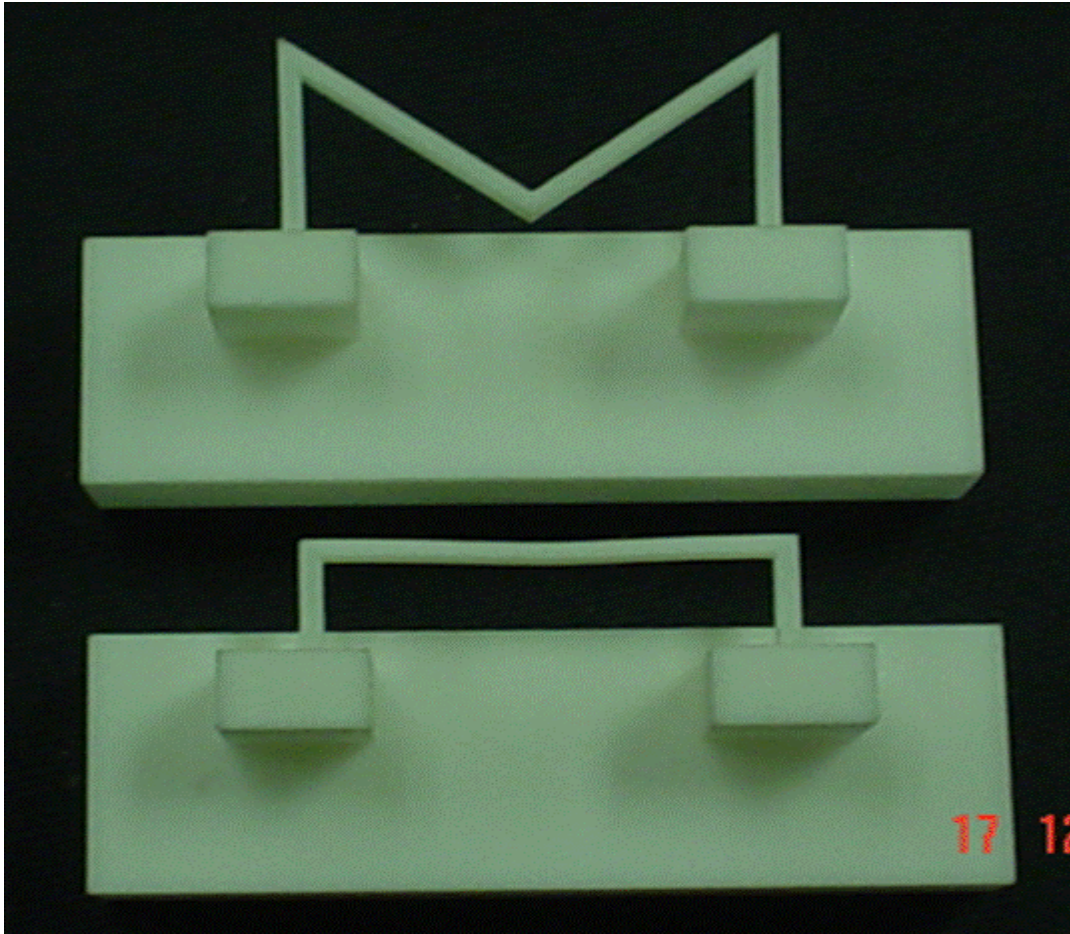


Figure 6.3 SLS manufactured “macro” prototypes

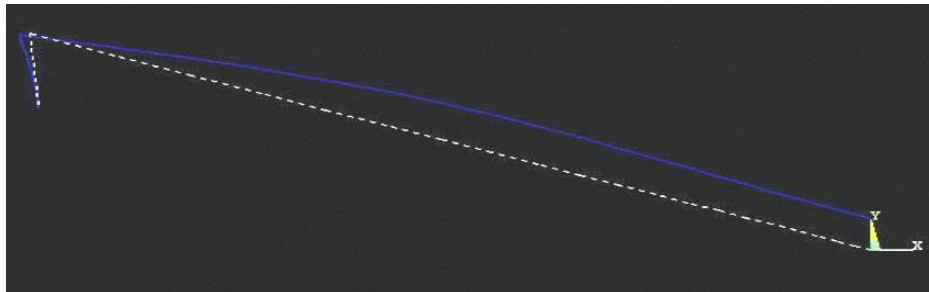


Figure 6.4 Feasible solution 1 generated in ANSYS by Q2S2 Methodology

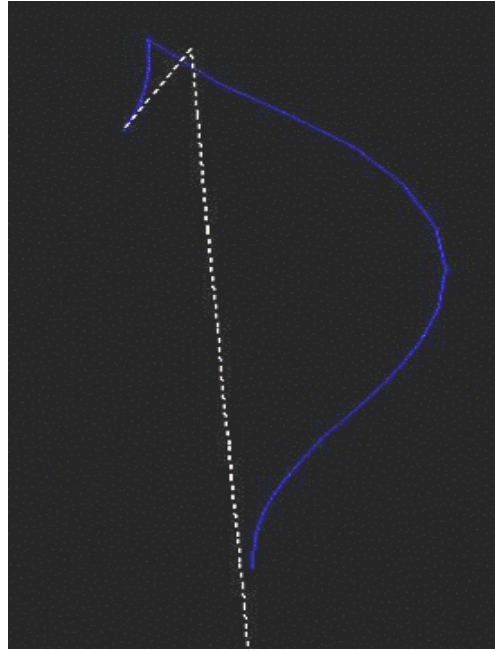


Figure 6.5 Feasible solution 2 generated in ANSYS by Q2S2 Methodology

(less than 10 times the number of variables). With these inputs Q2S2 was able to generate 2 feasible solutions (Figure 6.4 and 6.5). Previously, [Eashwer, 2004] it took on an average 3000 FEA evaluations to generate feasible solutions. However, the methodology presented in Eashwer et.al. [2004], is capable of producing more complex structure and works well for higher number of beams also.

6.3. Recapitulation

In summary, this example shows how the Q2S2 methodology can be used for mechanical design problems. Existing sampling techniques like the FF, LHD or HSS are not capable of doing this directly. The advantages of using qualitative information in conjunction with quantitative information through Q2S2 methodology opens up a whole

new avenue in which sampling techniques themselves can be used to not only provide design insights but also guide the overall design process. The implementation shown in this chapter exhibits possibility of wide applicability of the Q2S2 method over a range of system design problems.

This concludes the set of application problems in a satisfactory way. In the next chapter the major conclusions and future work that can be derived from this work are drawn.

Chapter 7

Conclusions

This chapter summarizes the dissertation, explains the support for the hypothesis and further draws some conclusions which have been reached during the process of completing this work. Finally, the possible extensions of the current work are also proposed.

In summary, a new qualitative and quantitative sequential sampling process for complex systems design has been formulated and shown to be theoretically as well as practically applicable to mechanical design problems such as a Bi-stable MEMS relays.

In addition to providing a mathematically sound and efficient method that provides good sampling points given the quantitative and qualitative information, it also furnishes an estimate of the *Confidence* we have about design space. This makes it possible to find sampling points that lead to higher accuracy in a far more advanced way than has been possible with existing sampling techniques. It also opens up new avenues to identify sampling points in interesting regions of design problem. Existing methods distribute sampling points to achieve statistical superiority without any heed to designer intent. Q2S2 provide sufficient degree of sophistication in incorporation of qualitative information depending on the intended purpose of the designer.

7.1. Summary of Results

In Chapter 3, the need for a methodology which can integrate both qualitative and quantitative information from varying fidelity sources is outlined. Different requirements for such a methodology is also presented in this chapter.

The fundamentals behind the Q2S2 methodology is presented in Chapter 4. Based on the requirements cited in Chapter 3 the idea of C-field was presented. How the C-field concept be used in sequential sampling in also presented in this chapter. The mathematical details behind the Q2S2 is elaborated in Chapter 4.

The function implementation in Chapter 5 is intended to demonstrate how the Q2S2 method works on known functions. It shows graphically and numerically how responses whose general behavior is not known are more likely to be better approximated by the Q2S2 method than by a existing sampling techniques for most cases. Also in this chapter the basic Q2S2 process is explained and demonstrated step by step at the example of a single variable function for visualization. Finally, in Chapter 6 it is described how the Q2S2 methodology can be used for mechanical design problems.

To the best of our knowledge this is the first attempt to include qualitative knowledge into sequential sampling techniques. The Q2S2 allows us to incorporate information of varying degrees of precision including detailed experiments, computational simulations, back-of-the-envelop calculations, and human intuition. The Q2S2 relying on human qualitative knowledge in conjugation with already existing quantitative techniques focuses efforts on important regions of the design space, thus eliminating extra sampling points and reducing the number of actual experiments performed. The novel strategy based on finding the location where we have the least information about the design problem works at least as well as existing methods and it eliminates computationally expensive processes like determining the covariance or Mean Squared Error (MSE) calculations. In all these examples, the Q2S2 method leads to higher accuracy metamodels than existing sampling techniques.

7.2. Reprise of Hypothesis

The hypothesis, formulated in Chapter 3, states that an alternative sampling process to the existing one can be formulated and used in its place. Further, it has been suggested that there are certain advantages to this choice, such as combining qualitative and quantitative information, and integrating information from varying fidelity sources. This aspect is a great enabler for the contributions of the sampling technique idea to complex systems design; according to the thoughts put forth in Chapter 1.

Specifically, the hypothesis states:

A sequential sampling process based on merging both quantitative and qualitative information coming from varying fidelity has the potential of producing higher accuracy models for design exploration purpose.

Four requirements were derived that need to be demonstrated, in order to provide support for this hypothesis:

1. Accuracy of prediction
2. Use in mechanical design domain
3. Ease of Application
4. Efficient use of Information

7.3. Conclusions on Why the Hypothesis Is Supported

To form a logical argument, the four points derived earlier are addressed:

1. The first point, the *accuracy* achieved through the sampling points predicted by Q2S2 was shown to be sufficiently higher than the existing sampling techniques when the number of sample points were the same.

The higher the number of variables in problem and lower the number of sampling points the better was the performance of Q2S2.

2. The use of Q2S2 methodology in *mechanical design domain* was demonstrated through the design of Bi-Stable MEMS relay. Through the relay example it was also demonstrated the Q2S2 methodology has the potential to accelerate design cycle and generate new design configurations.
3. Concerning *ease of application*, Chapter 5 outlines comprehensively how to apply the different steps so that this can be accomplish. A Matlab based program was written which interacts with user to implement different stages of Q2S2 process (Appendix A). Once the problem is defined, there is no special knowledge of the theory necessary, in order to successfully apply the process. This is accomplished through the selection of a various functions in different stages of Q2S2 process so that a well defined equations can be generated and optimized easily without detailed knowledge of Q2S2
4. The *efficient use of both qualitative and quantitative information* is demonstrated for 5 mutli-variable function problems, where it is shown that combining qualitative and quantitative information in coherent way leads to higher accuracy metamodels. The Bi-Stable MEMS relay design problem highlights the efficacy of merging qualitative and quantitative information in Chapter 6.

Having demonstrated these points in an empirical setting, the conclusion is reached that Q2S2 process can indeed be applied in a system-level integration design setting to model unknown design spaces. It also offers the designer valuable insight into

the problem, the nature of the design space, and the amount of information regarding this space, i.e. whether further samples are necessary, and if so, where. It is thus concluded that the hypothesis put forth earlier has been supported satisfactorily.

7.4. Suggestions for Future Research

There are several different directions in which to expand this research.

First, in the aspect of achieving a truly robust process for implementation, several steps need to be taken. Predominantly, a task would be to streamline and automate the optimization part of the Q2S2 process. This optimization can be quite sensitive, and the C^* surface (equation 4.36) is often multi-modal. Currently simple line search method from *fmincon* function in Matlab is used for this purpose. For this reason, it would be beneficial to employ some type of global optimizer, even if discrete, to at least find a good starting point at which the gradient search can begin. Furthermore, this type of optimization would provide an interesting application for a variety of heuristic search methods, many of which have developed in the last 10 years. Since information about good starting points (Peaks of C^* Field) is available, a parallel optimization process with *fmincon* function is implemented. However, using techniques like Parallel Simulated Annealing (PSA) may prove to be more beneficial.

Another area of research related to the concern for robust application would be the development of an intelligent front-end to pre-process the data. This could ensure that the data is scaled properly, and will not easily introduce numerical difficulties. In general, as long as a properly prepared data set is used, there are no requirements as to the structure of this data, i.e. no requirement for orthogonality; however if it is singular, there will be problems. A front end, which checks for this and prepares the data accordingly before it is passed to the process, would increase the usability in cases where it is not

generally known how to check for possible singularities or how to resolve a singularity if it occurs.

Recommendations affecting the method itself include the investigation of incorporating other types of qualitative information such as sensitivity information. There are various ways of representing qualitative information about a problem (references kuiper, etc). However, these ways of representing are not suitable for meaningful processing in Q2S2 framework. One idea would be to convert these representations by assuming proper mathematical forms for each of these representations. Once this is done every type of qualitative information can be integrated in Q2S2 methodology.

Another useful extension of this work could be creating a qualitative information repository. Various types of qualitative information available in design field (from first principles or otherwise) could be stored in this repository. Depending upon the nature of the design problem at hand, suitable qualitative information about the problem could be selected by a user. This can aid the not-so-experienced designer. Another use of such a repository would be using qualitative information from different design problem to explore the design space of a new design problem. For example, the idea would be to use qualitative information for the design of helical spring to design dual helix torsion spring.

One of the most useful extensions of this work could be changing the way the qualitative and quantitative information is stored in Q2S2. Currently, we have pre-defined mathematical models which are useful in storing qualitative and quantitative information. One method that can be useful is gaussian process. In the case of a gaussian process representation, there is no longer any model, or predefined mathematical forms. Rather, the information contained in the set of points is used to represent qualitative and quantitative information, and thus each new item of information is in essence a statistical estimate, rather than the mere an approximation model.

Whenever there is uncertainty regarding the true nature of the information, it is recommended to employ a gaussian process method, since it eliminates the need to assume a model, and simply gives a variance estimate. Processes similar to gaussian process but having a different way of representing statistical estimate could also be used. Another useful avenue could be using Fuzzy Theory [Zimmerman, 2001] based models to represent impreciseness in the qualitative information.

The current implementation of Q2S2 does not rely upon existing metamodeling techniques. However, it will be beneficial to develop a metamodeling technique that is customized to incorporate qualitative information. For example, if we know that in certain regions of \mathbf{x} there exists piecewise monotonicity (linear, quadratic so on) and directly map that information into the metamodel, it will be able to model the true nature of the problem more accurately. A qualitative and quantitative metamodeling technique customized to incorporate qualitative information would be very useful extension of this work.

Appendix

Software Manual

This Appendix is intended to give a basic understanding of the Matlab implementation of Q2S2. The comments in the code itself will provide more detail for advanced users. To receive a copy of the source code (available for academic use only), send an email to rahulrai@mail.utexas.edu listing your name, affiliation, and a brief synopsis of how you will use the code.

Framework and Important Files

Q2S2 requires a number of files to run. The master program, Q2S2, calls all the other subroutines needed including the function calling files created by the user, which will be referred to as `myfun` throughout this manual. The flow of information is summarized in Figure A.1. The solid boxes represent the various subroutines, with their respective names shown above. The gray boxes represent the directory where definition and methods to access various objects are stored.

The main code (Q2S2) begins with defining the problem. In order to define a problem a `Problem_Definition_obj` object is first instantiated. The relevant files for problem definition is stored in `@Problem_Definition_obj` directory. Once the problem is defined, various information available are created in object format. The `information_obj.m` helps us in storing information such as actual experiment information, simulation information or user guesses. The relevant files for information definition is stored in `@Information_obj` directory. Next step is the creation of monotonicity object with the help `Monotonicity_Information_obj.m`. The relevant files for monotonicity information definition is stored in

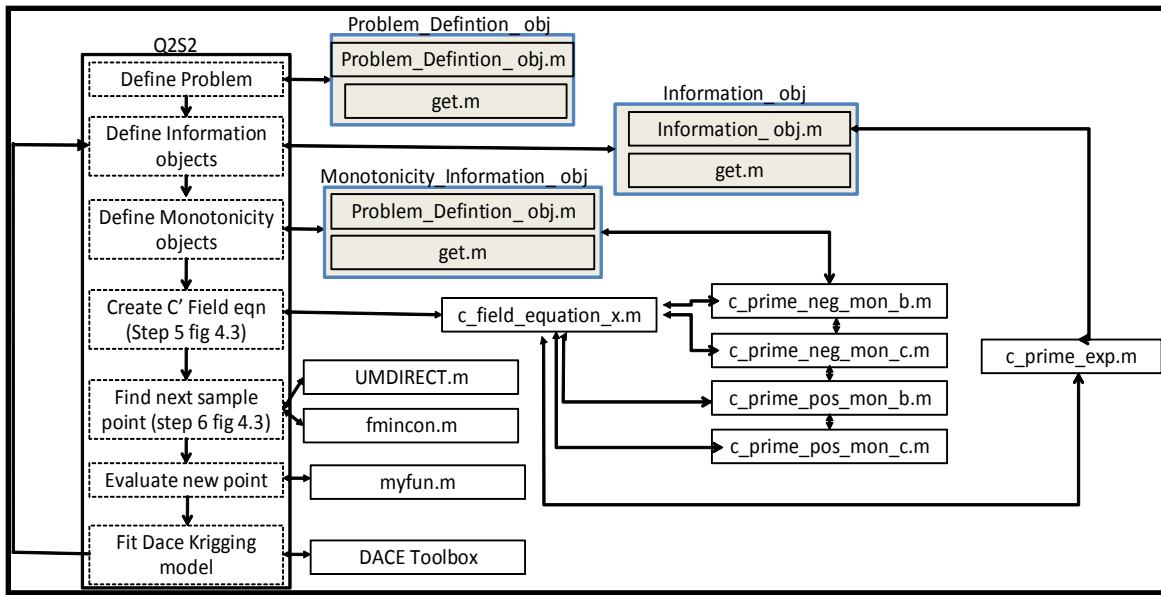


Figure A.1 Organization of MATLAB code for Q2S2

@Monotonicity_Information_obj directory. Once the problem is defined the C' field equation (at step 5 Figure 4.3) is derived by `c_field_equation_x.m`. The `c_field_equation_x.m` interact with various subroutines, `information_obj` and `monotonicity_information_obj` to derive the equation of the C' field.

The subroutines `c_prime_exp.m`, `c_prime_neg_mon_b.m`, `c_prime_neg_mon_c.m`, `c_prime_pos_mon_b.m`, and `c_prime_pos_mon_c.m` calculate constants A, B, and C defined in equation 4.29, 4.30, and 4.31. Once the C' field equation is constructed it is passed on to `UMDIRECT.m` or `fmincon.m` for locating the minima of C' field. This minima is the next sampling point. At this new sampling point a new evaluation of objective function is performed by using `myfun.m`. Finally, with all previous sample point and the new sample point a krigging metamodel is created by DACE toolbox.

The Subroutines

While most of the subroutines used in Q2S2 are only used internally, some may be used independently. Here we describe the basic functionality of those functions, their set of input and output arguments, and the portions of code that the user may want to change. Detailed descriptions of the programs can be found in the header and comments of the Matlab files themselves. All programs have been coded by Rahul Rai except DACE toolbox which was coded by Søren N. Lophaven, Hans Bruun Nielsen and Jacob Søndergaard, UMDIRECT which was coded by Ryan Fellini, Michael Sasena, and John Whitehead and `fmincon` which is part of Matlab's optimization toolbox.

MYFUNC

These are the function files that the user must create to compute the objective and/or constraint functions of the design problem.

UMDIRECT

This program is the University of Michigan version of Jones' DIRECT algorithm [reference]. It is a derivative-free method for constrained (or unconstrained), nonlinear optimization. DIRECT works by using Lipschitz theory to divide up the design space into hyper-rectangles. At each iteration, the set of hyper-rectangles most likely to contain good function values are further subdivided. The algorithm is used by Q2S2 to find the next sampling point by finding the minima of equation 4.36 for the problem at every iteration. The syntax for calling UMDIRECT is as follows (with optional input arguments shown in italics):

```
results=UMDIRECT(fileInfo,lb,ub,options,restartFile )
```

fileInfo information on the functions to be evaluated

lb lower bounds on design space

ub upper bounds on design space

options DIRECT running parameters

restartFile used to continue a previous DIRECT run

The first argument is a structure array with the following fields: `fName`, `gName`, `fParams`, `gParams`. The first two store the names of the function files that compute the objective and constraint functions, respectively. If the user would prefer to calculate both the objective and constraint functions in a single file, they must be assigned to `fName`. The `fParams` and `gParams` fields store any additional arguments that must be sent to the files `fName` and `gName` in order to execute them. The `lb` and `ub` arguments simply constrain the range of the design space. The `options` argument is another structure with the following fields:

`maxfCount` limit on function calls (default = $200 * (\# \text{ variables})$)

`maxIter` maximum number of iterations allowed (default = 50)

`termType` termination criterion (default = fCount OR Iter exceeded)

`conTol` allowable constraint violation (default = $1e-10$)

`display` amount of output sent to screen (default = show only final results)

`saveFile` save the results to file at each iteration (default = don't save)

`lgBalance` local / global balance parameter (default = $1e-4$)

`localSearch` allow for local optimization (default = use SQP)

Default values exist for all options, thus the user is not required to specify any of them. Several termination criteria exist, such as limiting the number of function evaluations or iterations, or stopping once very little progress has been made after a set

number of function evaluations or iterations. The reader is referred to the comments in the header of `UMDIRECT` for more details. The `lgBalance` option can be set by the user to control the balance between local and global searching. However, the default value has been shown by Jones to be quite robust, and therefore does not require tuning to the specific problem in most cases. The `localSearch` option allows the user to turn on or off the local search capability. While allowing for local searching is often quite helpful at efficiently locating the local optima, it may launch spurious local searches at times. If the function calls are inexpensive, the drawbacks are not significant, and local searching is recommended.

The final input argument, `restartFile`, is used to restart `UMDIRECT` in case the user would like to continue searching for a better solution beginning from the results of a previous run. It can be either the name of the `.mat` file containing the previous results, or the structure array of results itself.

The output of `UMDIRECT` is a structure array with the following fields:

<code>xBest</code>	best feasible design point found
<code>fBest</code>	objective function value at <code>xBest</code>
<code>gBest</code>	vector of constraint values at <code>xBest</code>
<code>fCount</code>	number of function evaluations performed
<code>iter</code>	number of iterations performed
<code>fChangeRate</code>	rate of change in <code>f</code> (used in <code>conWeight</code>)
<code>gChangeRate</code>	vector of rate changes for each constraint
<code>conWeight</code>	vector of constraint weights
<code>hist</code>	structure array containing history of best point

`rect` structure array of rectangle properties

`inputs` structure array of input arguments passed to `DIRECT`

`UMDIRECT` is a self-contained function. However, if its local search option is active, then it will call out to `fmincon` to find local solutions.

fmincon

This program is the Matlab SQP algorithm for constrained, non-linear optimization.

In some cases, it is called by Q2S2 to find the next sampling point by finding the minima of equation 4.36 for the problem at every iteration. The reader is referred to Matlab's help for more information on this program.

DACE Toolbox

This program is written by Informatics and Mathematical Modeling Group of Technical University of Denmark. For more detail please visit <http://www2.imm.dtu.dk/~hbn/dace/>.

```
[dmodel, perf] = dacefit(S, Y, regr, corr, theta0)
```

```
[dmodel, perf] = ...
```

```
dacefit(S, Y, regr, corr, theta0, lob, upb)
```

Input parameters:

`S` Design sites: an $m \times n$ array

`Y` $m \times q$ array with responses at `S`.

`regr` Handle to a regression function.

corr	Handle to a correlation function.
theta0	If lob and upb are not present, then theta0 should hold the correlation function parameters, μ . Otherwise theta0 should hold initial guess on μ .
lob,upb	Optional. If present, then they should be vectors of the same length as theta0 and should hold respectively lower and upper bounds on μ . If they are not present, then μ is given by the values in theta0.
Output:	
dmodel	DACE model. Struct with the elements
regr	handle to the regression function.
corr	handle to the correlation function.
theta	correlation function parameters.
beta	generalized least squares estimate.
gamma	correlation factors.
sigma2	estimate of the process variance.
S	scaled design sites, see (5.1) below.
Ssc	$2 \times n$ array with scaling factors for design sites.
Ysc	$2 \times q$ array with scaling factors for design responses.
C	Cholesky factor of correlation matrix.
Ft	decorrelated regression matrix.
G	matrix G .
perf	Information about the optimization.

Example Code for Projectile Problem

```
clear all;

% the problem definition, monotonicity information and initial user guess for the
projectile problem starts here

prob_def=Problem_Definition_obj(2,0,18,0,90,1,0,17);

max_q2s2_iteration=16 ;% maximum number of q2s2 iteration to be used or
sample %points to be used

% experimental or user guess begins below
experiments(1)=Information_obj([0 0],[2 10],0,5);
experiments(2)=Information_obj([18 90],[2 10],0,5);
experiments(3)=Information_obj([0 90],[2 10],0,5);
experiments(4)=Information_obj([18 0],[2 10],0,5);
experiments(5)=Information_obj([18 45],[2 10],17.65,5);

% monotonicity information begins below
mon(1)=Monotonocity_information_obj(1,2,0,30);
mon(2)=Monotonocity_information_obj(-1,2,60,90);

for j=1:max_q2s2_iteration
    for i=1:length(experiments)
        x(i,:)=get (experiments(i),'x_naught');% this part locates the peaks in overall
equation and are given as starting points during the optimization procedure

        %x2=[x1(1);x1(2)];

        feval(i)=get (experiments(i),'f_naught');

        OPTIONS=optimset('LargeScale','off');

        % Turn off all messages, even if no convergence.

        OPTIONS = optimset(OPTIONS,'display','off');
```

```

%GRAD=[c_field_equation_xgrad(x1,experiments,mon,prob_def);c_field_equation_xgr
ad(x1,experiments,mon,prob_def)];

%OPTIONS = optimset(OPTIONS,'gradobj','on','MaxFunEvals',200);

OPTIONS = optimset(OPTIONS,'MaxFunEvals',200);

%      grad      info      [x(i,:),fval(i),exitflag,output]      =
fminunc({c_field_equation_x(x(i,:),experiments,mon,prob_def),c_field_equation_xgrad(
x(i,:),experiments,mon,prob_def)},x(i,:),OPTIONS);

%no      grad      info      [x(i,:),fval(i),exitflag,output]      =
fminunc(c_field_equation_x(x(i,:),experiments,mon,prob_def),x(i,:),OPTIONS);

[x(i,:),fval(i),exitflag,output]      =
fmincon(c_field_equation_x(x(i,:),experiments,mon,prob_def),x(i,:),[],[],[],[],[0;0],[18;90
],[[]]);% this is optimization routine where equation is provided by c_field_equation_x
function

end

min_index=find(fval==min(fval));% this step find the index of peak which is
minima from among all the different peaks

xx=x(min_index(1),:)% this gives the next sample point

fvalue=fval(min_index(1));

% other attributes of new sample points are initialized here

new_exp_vector2 = [2 10];%input('input: x_tol:');

new_exp_vector3 =projectile(xx);%input('input: f_naught:');

% for black box function u can get this value by function call

new_exp_vector4 = 5;%input('input: f_tol:');

% finally the new sample point is stored as an information object in the next step.

```

```

        experiments(length(experiments)+1)
=Information_obj(xx,new_exp_vector2,new_exp_vector3,new_exp_vector4);
    end
    % Kriggin model fitting for all the sample points beings here
    for k=1:length(experiments)
        S(k,:)=get (experiments(k),'x_naught');
        Y(k,1)=get (experiments(k),'f_naught');
    end
    save data2.mat S Y;
    load data2.mat
    theta = [10 10]; lob = [1e-1 1e-1]; upb = [20 20];
    [dmodel, perf] =dacefit(S, Y, @regpoly0, @corrgauss, theta, lob, upb);
    X = gridsamp([0 0;18 90], 40);
    [YX MSE] = predictor(X, dmodel);
    avg_mse=sum(MSE)/length(MSE)
    % Kriggin model fitting for all the sample points ends here
    % plotting of the fitted model beings here
    %NOTE: only 1 or 2 variable problem can be plotted higher dimension can't
    %be plotted
    X1 = reshape(X(:,1),40,40);
    X2 = reshape(X(:,2),40,40);
    YX = reshape(YX, size(X1));
    figure(1), mesh(X1, X2, YX)
    hold on,

```

```
plot3(S(:,1),S(:,2),Y,'k', 'MarkerSize',15)% this part shows our sample points in  
the form of black dot in the figure
```

```
hold off
```

```
% plotting of the fitted model ends here
```

References

- Barton, R. (1992). "Metamodels for Simulation Input-Output Parameters," *Proceedings of the 1992 ACM Winter Simulation Conference*, Swain, J., et al, eds., Arlington, Virginia, December 13-16, 1992, pp. 289-99.
- Barton, R. (1994). "Metamodeling: A State-of-the-Art Review," *Proceedings of the 1994 ACM Winter Simulation Conference*, Tew, J.D., et al, eds., Orlando, Florida, December 11-14, 1994, pp. 237-44.
- Barton, R. (1998). "Simulation Metamodels," *Proceedings of the 1998 ACM Winter Simulation Conference*, Medeiros, D.J., et al, eds., Washington, D.C., December 13-16, 1998, pp. 167-74.
- Botev, Z. and Kroese, D. (2004). "Global Likelihood Optimization Via The Cross-Entropy Method With An Application To Mixture Models," *Proceedings of the 2004 ACM Winter Simulation Conference*, Ingalls, R.G., et al, eds., Washington D.C., December 5-8, 2004, pp. 529-35.
- Box, G.E.P. and J.S. Hunter., *The 2k-p fractional factorial designs*. Technometrics, 1961. **3**: p. 311-351.
- Burgess, C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition," *Knowledge Discovery and Data Mining*, **2**:2, pp. 1-43.
- Clarke, S., Greisbach, J., and Simpson, T., (2003). "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," DAC-48759, Proceedings of the 2003 ASME International Design Engineering Technical Conferences and Computers and Information in

Engineering Conference, Chicago, Illinois, September 2-6, 2003.

Cox, D., Turner, C., Legault, J., and Tesar, D. (1999). *Requirements for Small Automation Systems in Glove Box Environments*, ANRC Project 99-11 Report, The University of Texas at Austin, Austin, Texas, July 1999.

Craven, P. and Wahba, G. (1979). "Smoothing Noisy Data with Spline Functions: Estimating the Correct Degree of Smoothing by the Method of Generalized Cross-Validation," *Numerische Mathematik*, **31**, pp. 317-403.

DeVeaux, R., Psychogios, D., and Ungar, L. (1993). "A Comparison of Two Nonparametric Estimation Schemes: MARS and Neural Networks," *Computers in Chemical Engineering*, 17:8, pp. 819-37.

Dinh, Q., Turk, G., Yoo, T., Hart, J., O'Brien, J., and Whitaker, R. (2002). "Beyond Blobs: Recent Advances in Implicit Surfaces," 2002 ACM SIGGRAPH Conference Course Notes, Course #29, San Antonio, TX, July 21-26, 2002.

Drucker, H., Burges, C. Kaufman, L., Smola, A., and Vapnik, V. (1997). "Support Vector Regression Machines," *Advances in Neural Information Processing Systems 9*, Mozer, M., Jordan, M., and Petsche, T. eds., MIT Press, Cambridge, Massachusetts, pp. 155-61.

Dumais, S. (1998). "Using SVMs for Text Characterization," *IEEE Intelligent Systems*, July/August 1998, pp. 21-3.

Fan, H., Dulikravich, G. and Han, Z. (2004). "Aerodynamic data modeling using support vector machines," *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibition*, AIAA 2004-0280, Reno, Nevada,

January 5-8, 2004.

Fan, H., Dulikravich, G. and Han, Z. (2005). "Aerodynamic data modeling using support vector machines," *Inverse Problems in Science and Engineering*, **13**:3, pp. 261-78.

Frey, D.D., F. Engelhardt, and E.M. Greitzer, *A role for "one-factor-at-a-time" experimentation in parameter design*. Research in Engineering Design, 2003. **14**(2): p. 65-74.

Friedman, J. (1990). *Multivariate Adaptive Regression Splines*, Stanford Linear Accelerator Center Publication-4960, Technical Report 102, Stanford, California, August 1990.

Friedman, J. (1991a). "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, **19**:1, p. 1-67.

Gopi, M. and Manohar, S. (1997). "A Unified Architecture for the Computation of B-Spline Curves and Surfaces," *IEEE Transactions On Parallel and Distributed Systems*, **8**, pp. 1275-87.

Gurney, K. (2002). *An Introduction to Neural Networks*, Routledge, London, U.K..

Haykin, S. (1999). *Neural Networks: A Comprehensive Guide, 2nd Ed.*, Prentice-Hall, Upper Saddle River, New Jersey.

Hussain, M.F., Barton, R., and Joshi, S. (2002). "Metamodeling: Radial Basis Functions, versus Polynomials," *European Journal of Operations Research*, **138**, pp. 142-54.

Inselberg, A. (2002). "Multidimensional Visualization with Applications to Multivariate Problems," *2002 ACM SIGGRAPH Conference Course Notes*, Course #4, San Antonio, Texas, July 21-26, 2002.

Isaaks, E. and Srivastava, R. (1989). *An Introduction to Applied Geostatistics*, Oxford University Press, Oxford, U.K.

Jin, R., Chen, W., and Simpson, T. (2001). "Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria," *Journal of Structural Multidisciplinary Optimization*, **23**, p. 1-13.

Jin, R., Chen, W., and Sudjianto, A. (2002). "On Sequential Sampling for Global Metamodeling in Engineering Design," D A C -34092, *Proceedings of the 2002 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Montreal, Quebec, Canada, September 29-October 2, 2002.

Kalagnanam, J.R. and Diwekar, U.M. (1996), 'Robust design using an efficient sampling technique', *Computers & Chemical Engineering*, 20: S389-S394.

King, C., 2004 "Design Synthesis of Multistable Equilibrium Systems", Ph.D. Dissertation,
Mechanical Engineering Department, University of Texas at Austin, TX

Leijnen, J. and Sargent, R.G. (2000). "A Methodology for Fitting & Validating Metamodels in Simulation," *European Journal of Operations Research*, **120**, pp. 14-29.

Kollata, E., C.W. King, and M.I. Campbell. *Design Synthesis of Multistable Compliant Structures* in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2004. Albany, NY.

Kreinovich, V. (1991). "Arbitrary Nonlinearity Is Sufficient to Represent All Functions by Neural Networks: A Theorem," *Neural Networks*, **4**, pp. 381-3.

Kushner, H. (1964). "A New Method of locating the Maximum of an Arbitrary Multi-Peak Curve in the Presence of Noise," *Journal of Basic Engineering*, **86**, pp. 97-106.

Laslett, G. (1994). "Kriging and Splines: An Empirical Comparison of Their Predictive Performance in Some Applications," *Journal of the American Statistical Association*, **89**:426, pp. 391-400.

Locatelli, M. (1997). "Bayesian Algorithms for One-Dimensional Global Optimization," *Journal of Global Optimization*, **10**, pp. 57-76.

Manasararian, O. (2005). "Support Vector Machine Classification Via Parameterless Robust Nonlinear Programming," *Optimization Methods and Software*, **20**:1, pp. 115-26.

Martin, J. and Simpson, T. (2002). "Use of Adaptive Metamodeling for Design Optimization," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, September 4-6, 2002.

Mehrotra, K., Mohan, C., and Ranka, S. (2000). *Elements of Artificial Neural Networks*, MIT Press, Cambridge, Massachusetts.

Mockus, J. (1989). *Bayesian Approach to Global Optimization*, Kluwer Academic Publishers, New York, New York.

Mullur, A. and Messac, A. (2004). "Extended Radial Basis Functions: More Flexible and Effective Metamodeling," *10th Proceedings of the 2004 Multidisciplinary Analysis & Optimization Conference*, Albany, New York, August 30-September 1, 2004.

Mullur, A. and Messac, A. (2005). "Extended Radial Basis Functions for Metamodeling: A Comparative Study," D AC -85041, *Proceedings of the 2005 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Long Beach, California, September 24-8, 2005.

Myers, R.H. and Montgomery, D. (2002). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 2nd Ed., John Wiley and Sons, Inc., New York, New York.

Nicolai, R., Dekker, R., Piersma, N., and van Oortmarssen, G. (2004). "Automated Response Surface Methodology for Stochastic Optimization Models with Unknown Variance," *Proceedings of the 2004 ACM Winter Simulation Conference*, Ingalls, R.G., et al, eds., Washington D.C., December 5-8, 2004, pp. 491-9.

Osuna, E. (1998). "Applying SVMs to Face Detection," *IEEE Intelligent Systems*, July/August 1998, pp. 23-6.

Palmer, K. and K.-L. Tsui, *A Minimum Bias Latin Hypercube Design*. IIE Transactions, 2001. **33**(9): p. 793-808.

P.Y. Papalambros and D.J. Wilde. *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, New York, second edition, 2000.

Piegl, L. (1991). "On NURBS: A Survey," *IEEE Computer Graphics and Applications*, pp. 55-71.

Piegl, L. and Tiller, W. (1997a). *The NURBS Book*, 2nd Ed., Springer-Verlag, Berlin, Germany.

Platt, J. (1998). "How to implement SVMs," *IEEE Intelligent Systems*, July/August 1998, pp. 26-8.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (2002). *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd Ed., Cambridge University Press, Cambridge, U.K.

Rogers, D. and Adams, J. (1990). *Mathematical Elements of Computer Graphics*, 2nd Ed., McGraw-Hill, New York, New York.

Salford Systems. (2001). *MARSTM User Guide*, Salford Systems, San Diego, California, 2001.

Saloman, D. (1999). *Computer Graphics and Geometric Modeling*, Springer-Verlag, Berlin, Germany.

Sasena, M. (1998). *Optimization of Computer Simulations via Smoothing Splines and Kriging Metamodels*, Masters Thesis, Department of Mechanical Engineering, The University of Michigan, Ann Arbor, Michigan, May 1998.

Sasena, M., Parkinson, M., Goovaerts, P, and Papalambros, P. (2002a)

“A daptive Experim ental D esign A pplied to an E rgonom ics Testing Procedure,” D A C -34091, *Proceedings of the 2002 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Montreal, Quebec, Canada, September 29-October 2, 2002.

Sasena, M. (2002b). *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, Ph.D. Dissertation, Department of Mechanical Engineering, The University of Michigan, Ann Arbor, Michigan, May 2002.

Schölkopf, B . (1998). “S V M s – a Practical Consequence of Learning Theory,” *IEEE Intelligent Systems*, July/August 1998, pp. 18-21.

Schölkopf, B., Burges, C., and Smola, A. (1999a). *Advances in Kernal Methods – Support Vector Learning*, MIT Press, Cambridge, Massachusettes.

Schölkopf, B . (1999b). “Input Space V s. Feature Space in Kernal-Based M ethods,” *IEEE Transactions on Neural Networks*, **10**:5, pp. 1000-17.

Schonlau, M. (1997). *Computer Experiments and Global Optimization*, Ph.D. Dissertation, Department of Mechanical Engineering, The University of Waterloo, Waterloo, Ontario, Canada, 1997.

Simpson, T., Peplinski, J., K och , P . and A llen , J. (1997). “O n the U se of Statistics in Design and the Implication for Deterministic Computer Experim ents,” D T M -3881, *Proceedings of the 1997 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Sacramento, California,

September 14-17, 1997, pp. 1-12.

Simpson, T. (1998). *Comparison of Response Surface and Kriging Models in Multidisciplinary Design of an Aerospoke Nozzle*, NASA/CR-1998-206935, ICASE Report No. 1998-16, NASA Langley Research Center, Hampton, Virginia.

Simpson, T., Peplinski, J., Koch, P., and Allen, J. (2001). "Metamodels for Computer-based Engineering Design: Survey and Recommendations," *Engineering with Computers*, **17**, pp. 129-50.

Taylor, H. W., Brand, O., Allen, M. G., 1998, "Fully integrated magnetically actuated micro machined relays", *JMEMS* **7**, 181-191

Turner, C. (2002a). *Hyperdimensional Performance Models (HyPerModels) for Engineering Design: A Ph.D. Proposal to The Faculty of The University of Texas at Austin*, Doctoral Committee Proposal, Austin, Texas, December 2002, LA-UR-03-0982.

Turner, C. (2002b). "Metamodels for Planar 3R Workspace Optimization," CIE-34500, *Proceedings of the 2002 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Montreal, Quebec, Canada, September 29-October 2, 2002.

Turner, C. (2005b). "Determining Experimental Adequacy with Adaptive Data Collection and Metamodels," Los Alamos National Laboratory Unclassified Release to the 2005 University of Texas Graduate Engineering Research Conference, Austin, Texas, April 28, 2005, LAUR-05-2791.

Van Beers, W. and Kleijnen, J. (2004). "Kriging Interpolation in Simulation: A Survey," *Proceedings of the 2004 ACM Winter Simulation Conference*, Ingalls, R.G., et al, eds., Washington D.C., December 5-8, 2004, pp. 113-21.

Van Gestal, T., Suykens, J., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B., and Vandewalle, J. (2002). "Benchmarking Least Squares Support Vector Machine Classifiers," *Machine Learning*, in Press.

Wang, N. and Ge, P. (1999a). "Study of Metamodeling Techniques and Their Applications in Engineering Design," *ASME-MED Manufacturing Science and Engineering*, **10**, pp. 89-95.

Wang, T. and Wu, K. (1999b). "A Parallel Set Design Procedure for the Simulated Annealing Algorithm Under Computational Time Constraints," *Computers and Operations Research*, **26**, pp. 665-78.

Watson, A. and Barnes, R. (1995). "Infill Sampling Criteria to Locate Extremes," *Mathematical Geology*, **27**, pp. 589-608.

Vita

Rahul Rai was born in Rohtas, Bihar, India on March 1, 1978 to Vidya and Shanti Rai. After initial schooling from Bhola Nanda National Vidyalaya, Calcutta, India, he entered the University of Ranchi, India, majoring in Manufacturing Engineering. While a student at the University of Ranchi, Rahul published 3 journal papers and presented 3 papers at international conferences. Rahul graduated from the University of Ranchi with honors in May 2000. After his Bachelors he entered the University of Missouri-Rolla in June 2002 in the Sustainable Design Lab earning a Master of Science in Manufacturing Engineering in May 2002. Interest in design methodology motivated Rahul to pursue a Ph.D. in Mechanical Engineering at the University of Texas at Austin. Since 2002, Rahul worked with the Automated Design Laboratory. Till Date Rahul is lead author to more than 15 journal and conference papers. Rahul completed his Ph.D. in summer of 2006.

Permanent address:

174/13A/3C, New Mehdauri Colony, Teliarganj, Allahabad, UP, India-211004

This dissertation was typed by the author.